AUTONOMOUS UAS CONTROLLED BY ONBOARD SMARTPHONE

A Thesis Presented to
The Faculty of the Computer Science Program
California State University Channel Islands

In (Partial) Fulfillment
of the Requirements for the Degree
Masters of Science in Computer Science

by
Phillip Gabriel Bryant
December 1st, 2016

APPROVED FOR THE COMPUTER SCIENCE PROGRAM

12/01/2016

Advisor: Dr. David Claveau — Date

12/01/16

Dr. Michael Soltys — Date

12/01/2016

Dr. Jason Isaacs — Date

APPROVED FOR THE UNIVERSITY

12-1-16

Dr. Gary Berg — Date

Autonomous UAS Controlled by Onboard Smartphone

by

Phillip Gabriel Bryant

Computer Science Program

California State University Channel Islands

Abstract

This thesis presents an autonomous unmanned aircraft system (UAS) that is controlled entirely by an onboard consumer smartphone. The smartphone has become an affordable and easy-to-use device for the average person. It is typically designed to be lightweight, energy-efficient, and has impressive computational resources with powerful sensors giving it the potential to serve as a controller for an autonomous system. To fully control a UAS, however, there are several challenges. Some of these involve low-level flight control which includes the need to sense the current state of the UAS and actuate flight-control surfaces in real time. Other challenges involve the navigation of the UAS in order to perform some useful behavior. The typical solution is to use custom control hardware and software that are often expensive and not accessible to the average person. This thesis shows how a single smartphone can, based upon user-supplied GPS flight path coordinates, subsequently take flight, collect relevant data, and safely land without further instruction. This project consists of a 1.5-meter wingspan airplane carrying an Android smartphone connected via IOIO interface to servos that actuate the control surfaces and motor. With the exception of the initial GPS coordinates, no live direction is necessary for operation. Primarily, this level of autonomy was achieved by aggregating a diverse set of known algorithms to collectively validate the feasibility of smartphone-controlled autonomous systems.

Keywords—Autonomous UAS, Android, Autopilot, IOIO, Onboard Smartphone

Acknowledgements

This project would be impossible without the extraordinary support of my wife, Jenessa, whose boundless encouragement transcended the disruptive and egregious schedule. The immense professional talent of Greg Gradwell influenced all of the content here, in many ways I feel as though I was simply the reporter as we worked together to complete the many trials herein. And, of course, my advisor, Professor David Claveau, who calmly and readily provided content and counsel throughout the entire process and steered me toward excellence.

TABLE OF CONTENTS

TABLE OF FIGURES

KEY TERMS

AGL – Above Ground Level

Autopilot – A computational system that flies, or drives, a vehicle

COTS – Commercial Off the Shelf

Drone – A generic term for robotic vehicles, generally airborne

GPS – Global Positioning System

GUI – Graphical User Interface

I2C – Inter-Integrated Circuit (technically I^2C)

IOIO – Pronounced "yo-yo", is an open source Android microcontroller

PID – Proportional, Integral, Derivative control system

PWM – Pulse Width Modulation

RC – Remote Control

RPV – Remotely Piloted Vehicle

UAS – Unmanned Aerial System

UAV – Unmanned Aerial Vehicle

Chapter 1: Introduction

1.1 Introduction to the Proposed System

Autonomous robots are beginning to change a wide range of industries such as transportation, manufacturing, and healthcare. Perhaps the most prominent of these autonomous robots are those designed for flight, the ones which have been colloquially referred to as drones. Researchers and standards bodies usually refer to a drone as an Unmanned Aerial Vehicle (UAV) or an Unmanned Aircraft System (UAS). In the near future, these UASs are expected to be used in ways that will improve crop yields, reduce fire and crime risks, and improve shipping times.[1] This type of aircraft has no onboard crew or passengers. While definitions can vary, it is generally accepted that for an aircraft to be properly categorized as a UAS, it must have a system called an autopilot which is capable of controlling all aspects of the aircraft's flight. This includes high-level control such as navigation by visiting pre-defined waypoints. It also includes low-level control such as maintaining level flight and handling minor perturbations such as low-speed cross winds. It should also be capable of takeoff and landing and should be interruptible

by a remote human operator. These requirements are usually met by using specialized real-time hardware and software that are often expensive and difficult to use for the average person. This thesis shows how a commercial, off-the-shelf smartphone can be used to satisfy the requirements of a UAS autopilot while remaining accessible to the average user. This is expected to lead to even more applications of the already popular UAS technology.

The next section provides some historical background to help the reader understand and situate the work presented in this thesis. It may be skipped if the reader is already familiar with this history.

1.2 Historical Context

Unmanned aircraft systems have been used since at least the middle of the 19th century when Austria employed balloon-borne ordnance against Venice, Figure 1.[2] At the end of that century, in 1898, Nikola Tesla demonstrated vehicle remote control, dubbed 'teleautomata', on a pond in Madison Square Garden. Tesla's patent was one of the first applications of radio waves in history and was ushered in alongside a new century of flight. A couple decades of rapid innovation passed, and in the waning years of WW1, the Ruston Proctor Aerial Target and the Kettering 'Bug', shown in Figure 2, were borne. By combining teleautomata and intentional isolated ordnance, these remotely piloted vehicles (RPV), somewhat

akin to a cruise missile, were designed to utilize radio control techniques to safely destroy Zeppelins from range.[3]

Figure 1. Austrian balloon borne ordnance[4]

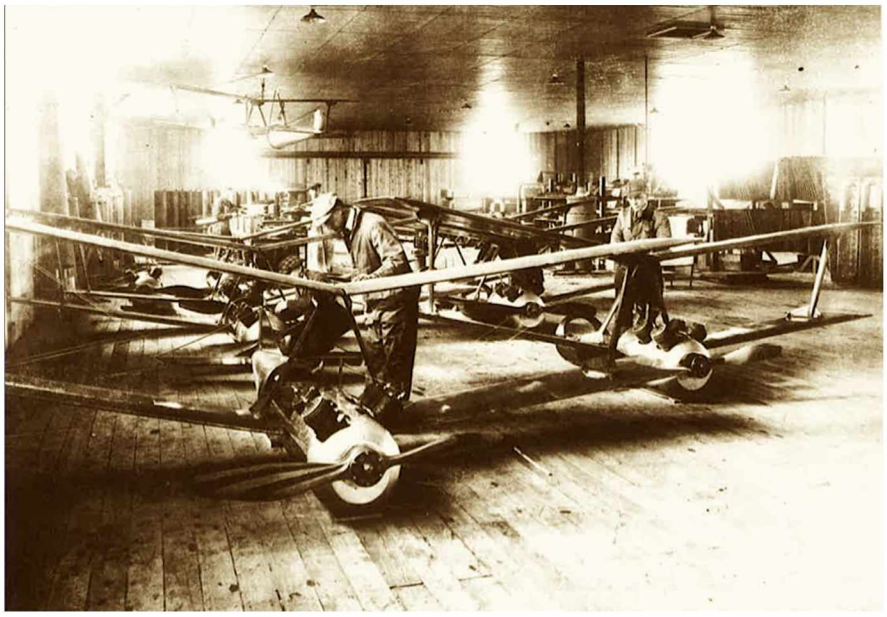


Figure 2. Kettering Bug remotely piloted vehicles[5]

Then, in the dawning years of World War 2, the term 'drone' was finally coined by the Officer in Charge of the US Naval Research Laboratory as a

descriptive of the radio-controlled aerial targets under development in 1936. Around this time, a British immigrant named Reginald Denny opened a shop in the greater Los Angeles area that would soon produce RPVs by the thousands. Simultaneously, television-assisted torpedoes dubbed "Project Fox" that contained an onboard RCA camera for video capture were underway. The plan was to retrofit a fleet of fewer than 200 aircraft controlling more than 5 times their number in drones.[6] Radio-controlled drones became a hallmark of military home arsenal; Reginald's shop was soon the Radioplane Company (from whence Marilyn Monroe found her fame).[7]

Several more B-17 Flying Fortresses were outfitted for remote flight in the years leading up to the 1950s. Takeoff and landing were commanded from a Jeep on the ground while in flight, a sister B-17 maintained control. The remote aircraft were most notably used to collect data and samples from the radioactive cloud produced by atomic projects: Operations Crossroads (Bikini Atoll), Sandstone, and Greenhouse.[8] Not long after, as several local residents to Channel Islands still remember, Pt. Mugu was being used as a test base for RPVs. On one particular morning in August of 1956, a Grumman F6F-5K drone took flight under remote operation. Soon, however, it stopped responding to commands, heading toward Los Angeles. Two F-89D Scorpions were dispatched to bring down the drone, but were unable to land a direct hit before running out of fuel, with stray missiles

bouncing around the streets, starting fires, and damaging property. Finally, the F6F-5K crashed a few miles east of the Palmdale Regional Airport. There were no fatalities: ever since locals have referred to the day as the Battle of Palmdale.

These developments morphed into the idea of remote reconnaissance during the Vietnam War. The demand for an RPV capable of penetrating deep into enemy territory escalated dramatically in 1960 when a U-2 pilot, Francis Gary Powers, was shot down over the USSR.[9] The highly-classified Red Wagon RPV program was born; tasked to modify the Ryan Firebee targets, Figure 3, which were previously used to verify anti-aircraft systems.[10] Just a few years later, in 1964, with the "Tonkin Gulf Resolution" and the birth of the Vietnam war, the first Ryan 147Bs (Lightning Bugs) were deployed from Hercules C-130 gunships. The Lightning Bugs carried onboard cameras, flew a preprogrammed route taking still photographs for up to two hours, and were collected by helicopter after they parachuted for recovery near Taiwan.

Over the course of the Vietnam War, nearly 3500 Ryan reconnaissance drones were flown.[9] The majority of these target drones were launched without landing gear from aircraft or off a rail using solid-fuel rocket-assisted takeoff (RATO) boosters, as shown in Figure 3. In the 1970s, the Air Force launched the Compass Cope program to increase the range and surveillance capabilities of RPVs. The prototypes were potentially capable of flying for over 24 hours while

commanded by Ground Control Stations (GCS). DARPA increased funding with the Remotely Piloted Aerial Observation/Designation System program in 1972. After a series of forerunners, tests, and requests for proposal (RFPs), eventually Lockheed's MQM-105 Aquila took its first flight in 1983. The Aquila was launched hydraulically and controlled from a 5-ton truck. Its target acquisition, designation, and aerial reconnaissance (TADAR) fulfillment package was a Westinghouse television image/laser designator, a secure communications link (radio with protocols), and a Near IR detector.[11]

Figure 3. Ryan Firebee rocket assisted takeoff[12]

While the American programs labored under budget cuts and waned under the general disapproval of the public eye, the Israeli Air Force was moving full steam ahead. By the 1970s, the brilliant engineer Abraham Karem was building

drones for the Israeli Air Force. In 1982, Israel coordinated the use of these UAS as decoys, jammers, and instant video recon alongside manned aircraft. The result was a swift victory over a Syrian fleet. In the same year as this attack, Karem was convinced of the technological prowess of drones. He moved to America and built a cigarish aircraft dubbed the 'Albatross' which could fly for 56 continuous hours, Figure 4. With pressure from the impressive Israel engagement of UAS, progress was rapid. Then political upheaval, including a congressional block of all UAS funding, led to General Atomics assimilating Karem's team, and Albatross became GNAT-750. The GNAT was equipped with GPS navigation, infrared low-light cameras in a moveable nose sensor turret, and C-band datalink of up to 150 nautical miles. It famously stayed funded only by a then-secret CIA project called Lofty View during the Bosnian war.[13]

The GNAT was still very limited. It performed poorly in weather and central control was, at best, an email exchange from Langley to the operator in Albania. Thus, the Predator was born: essentially a GNAT with laser-guided targeting, de-icing, and, most importantly, a SATCOM datalink allowing worldwide communication. It took another 7 years of development before it made its first strike, and the program to build and maintain these long-range surveillance UAS has cost nearly $3 billion to date.[14] This program even included the allocation of the Creech Air Base, designed exclusively to support UAS operations.

Figure 4. Karem's Albatross, predator forerunner[15]

Within the last decade, spurred by the extensive military investment--$2.9 billion FY16, the civilian marketplace has undergone a grassroots revolution.[16] Large companies have continued development. For example, in 2007 Raytheon worked with the University of North Dakota to build a 10-foot wingspan drone, dubbed Cobra, to test agricultural and civilian surveillance applications.[17] But startups and homebrew autopilots trickling into the marketplace ripe with hobbyists has driven the market. In the West, in 2009, 3DR Robotics built the first widespread, affordable autopilot with Ardupilot, Figure 5. Around the same time, DJI Robotics in China was delivering their hobbyist helicopter autopilot, the Innovations XP 3.1. These effective, inexpensive solutions have stirred demand, with the commercial industry expected to generate $12 billion in sales in the ensuing decade.[18]

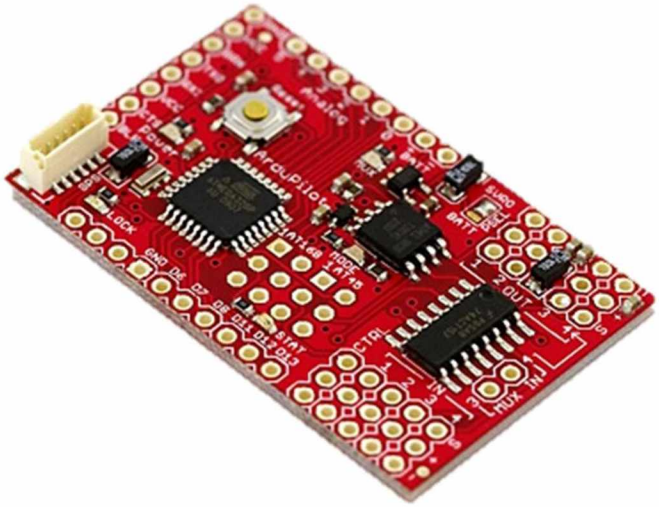


Figure 5. Ardupilot, an open-source autopilot platform[19]

1.3 The Proposed UAS

Today, the UAS industry is a collection of designs ranging in size from the human hand to a semi-truck. Two designs that are indicative of this broad range are AeroVironment's Puma[20] and DJI's Phantom[21], Figures 6 and 7 respectively. The Phantom is a quadcopter built primarily to support hobbyist cinematography. It is semi-autonomous with sixteen programmable waypoints and automatic landing. It costs approximately one thousand US dollars and has a flight time of twenty-five minutes.[22] The Puma is a hand-launched aircraft that focuses on intelligence gathering for military applications. It has both manual and autonomous control, can stay airborne for over three hours and nine miles, and costs about one quarter of a million dollars.[23] Cost and knowledge barriers eliminate a large part of the

population from the potential user base for these systems and the benefits they can provide, while affordable and simple systems are constrained in terms of operational effectiveness. The UAS presented in this thesis represents a solution between the two. It is intended to be accessible to the general public while offering the sophisticated autonomy of an advanced UAS. It does this by leveraging the surprising capabilities of today's smartphone technology.

Perhaps the first successful commercial smartphone was the iPhone which was introduced by Apple in 2007.[24] In the nine years since its release, the smartphone market has pervaded the planet. Over a billion units are sold each year, more than half of which are on the competing Android platform.[25] They are now an essential part of modern society. They are designed to be easy-to-use, and come equipped with advanced sensors and processors that surpass those available on a typical consumer-grade UAS autopilot. Because the average person has become familiar with smartphone operation it was expected that a smartphone-based autopilot could be user-friendly and affordable without sacrificing essential UAS capabilities. There have been several similar attempts at this design approach[26,27,28], but none demonstrate fully autonomous flight.

Figure 6. AeroVironment Puma[20]

Figure 7. DJI Phantom 3[21]

The primary challenge in this thesis was to design and build an autonomous system that could reliably replace the more expensive, complicated, and real-time offerings currently available. The three key aspects to improve were identified as cost, usability, and utility. Cost and usability were inherently solved by the smartphone itself, as they are nearly ubiquitous with applications that are very familiar to the layman. As smartphone technology advances, the sensor and software functionality, and therefore utility, continues to rise. Synchronized with this growing utility, the development cycles on a highly supported smartphone platform are notably shorter. Thus, the main challenge was to design and build the software that would allow the combination of the smartphone with a UAS. The remainder of this document describes how this challenge was met.

1.4 Remaining Chapters

In the second chapter, a survey of related work from antiquity to contemporary contributions is presented. The third chapter discusses the background of core technologies that were used to develop the proposed UAS. This includes a survey of the core algorithms and methodologies there were needed to achieve autonomy, alongside the first stages of automated control. These first stages include early experiments with a remote-controlled car that was commanded to drive around a course. The fourth chapter outlines how the proposed UAS was

designed, including the architecture and physical structure. The fifth chapter delineates the performance results of the proposed UAS, how successfully it managed flight, and several of the pitfalls encountered along the way. The sixth chapter looks at a selection of additional research which was completed during this thesis: a collection of stubs ripe for future work. The seventh and final chapter includes a summary of the project's results. The final chapter also compares different learning techniques, including what remains unsolved and what could be added to further enhance the project.

Chapter 2: Related Work

This chapter provides a historical background aerial systems control as it relates to this thesis. The chapter also surveys some of the more recent work done in this field and compares it with the approach taken here.

2.1 Early and Classical Control

One of the first mathematical models for stability was completed in the 1870s by Vyshnegradskii,

$$\varphi^3 + x\varphi^2 + y\varphi + 1 = 0, \quad (1)$$

who found that a point in the $x - y$ plane delineated the system transient response—that is the response a system to a change in equilibrium.[29] By the beginning of the 20th centuries, early textbooks were being published on automated control, including key details like the Hurwitz stability criterion that provides a construct to identify if linear equations of motion have stable solutions.[30] Within the ensuing century after the Wright brothers' historic 1903 flight, this nascent research would transform into ever-improving autopilots.

Wilbur Wright, in 1901, when speaking before the Western Society of Engineers, said, "Men already know how to construct wings or aeroplanes… [but] inability to balance and steer still confronts students of the flying problem… when this one feature has been worked out, the age of flying machines will have arrived."[31] This historic comment is still relevant today and is playing out in the field of practical, fully-automated flight, the focus of this thesis.

2.1.1 Sperry's Gyroscopic Stabilizer

In June of 1914, Lawrence Sperry was able to fly his aircraft while standing on one of its wings while his business partner, Emil Cachin, sat on the other wing—with an empty cockpit.[32] The moment prior to him walking onto the wing was captured in Figure 8. To the amazement of the crowd, the aircraft was held stable by a quartet of air-actuated hydraulic servo valves and gyroscopes. They interconnected to relatively detect the position of the aircraft and apply corrective action to the rudder, elevator, and ailerons.[33] The gyros themselves were an extension of Lawrence's father Elmer Sperry's work. They were essentially devices that maintained their orientation in space while the gimbals rotated around them as shown in Figure 9. What was particularly impressive about Sperry's gyroscopic stabilizer was that it included an intuitive incorporation of the derivative action in addition to an adaptive behavior, since the gain kept up with

the speed of the aircraft. Sperry invented this brilliant control system without the mathematical theory behind these insights, which were uncovered later.

Figure 8. Sperry flies without hands, with Emile Cachin on lower wing[34]

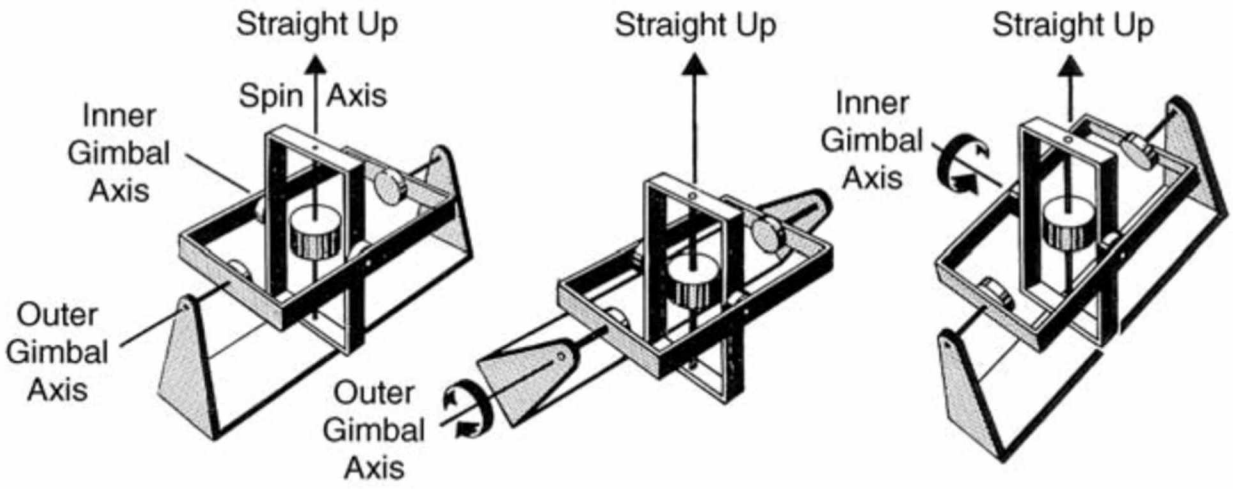


Figure 9. Displacement gyro, mounted on gimbals, maintains its orientation[35]

2.1.2 Directional Stability of Automatically-Steered Bodies

The excitement surrounding Sperry's device and WWI ensured significant technological investment in autopilot design. By the mid-1930s, a number of aircraft were using them for long-distance flight records. During that time, the

Russian scientist, Nicholas Minorsky, completed and published an important study of control.[36] Minorsky's researched was focused on the goal of general stability, not perfect control and was based on the observations of the actions of a helmsman who commands a ship based upon current error, previous error, and the rate of change between the two. His paper established a mathematical model upon which the pioneers of the aviation industry were largely building via intuition and trial and error.

Minorsky first argued that the requirements for good steering were known: "An efficient helmsman keeps the ship accurately on her course by exerting a properly timed *meeting* and *easing* on the rudder."[36] He went on to contend that a particular timing of the rudder in conjunction with current inertia would be sufficient to "reach the best possible conditions for directional stability of the body to be steered on its course."[36] This leads directly to the equation,

$$A\ddot{\alpha} + B\dot{\alpha} + k\rho = D, \tag{2}$$

where α is the ship's deviation angle from the desired course, ρ is the angle of the rudder, A is the ship's moment of inertia about the vertical axis, B is the ship's frictional resistance to turning, D is the disturbing torque, and k is a constant dependent upon the rudder's characteristics. Therefore, the rudder angle can be solved as a function of the deviation and its derivatives. After considering several

variations upon the conditions of this equation, he considers a controller of the form,

$$\rho = \int [m\alpha(t - T_1)]dt + n\alpha(t - T_2) + p\dot{\alpha}(t - T_3), \qquad (3)$$

and approximates based upon Taylor's expansion, assuming T_1, T_2, and T_3 are small, to obtain a series of conditions for stability. Essentially making the determination that a series of constants, *m, n,* and *p* here, determine the intensity while the integral provides the moderation of control. The limitation was "that all possible methods of rudder control *do not actually anticipate* the disturbing angular motion";[36] that is, they lacked a derivative control.

Minorsky continued to refine this idea, and by 1930, he had developed a method to combine the signals from the gyrometer, gyrocompass, acceleration, and rudder position (Figure 10). The two gyros were connected to motors that turned according to the angular position of the ship and angular velocity of the yaw (compass for position, meter for velocity). These twin signals combined to a differential gear that set the potentiometer (1). Acceleration control was loosely coupled with gyrometer (6) and two booster generators (8 and 8'). The potential, or amount, of boost was set by potentiometers 10 and 10'. In all, Minorsky's controller exhibited derivative and proportional control but was never adopted by the military.[37]

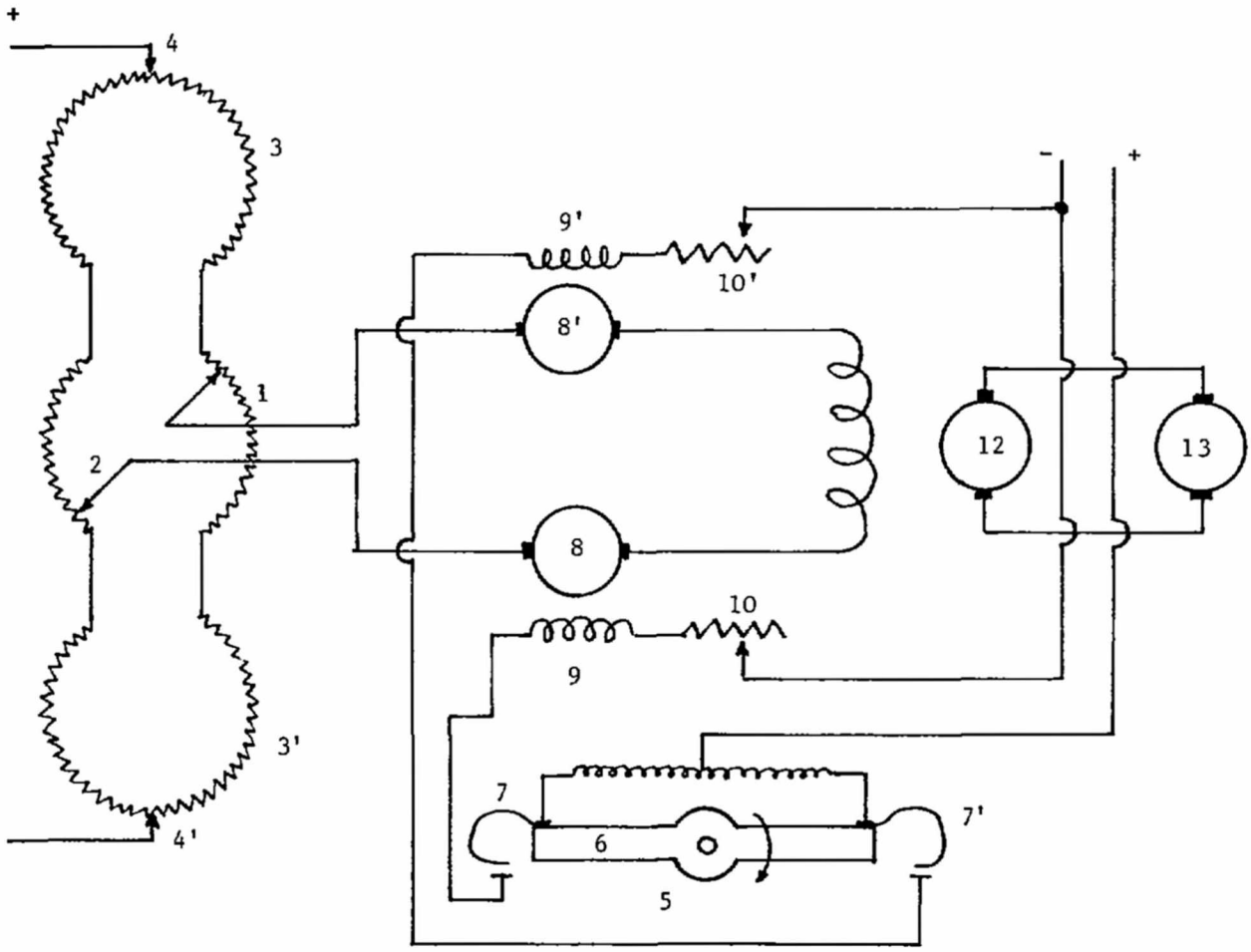
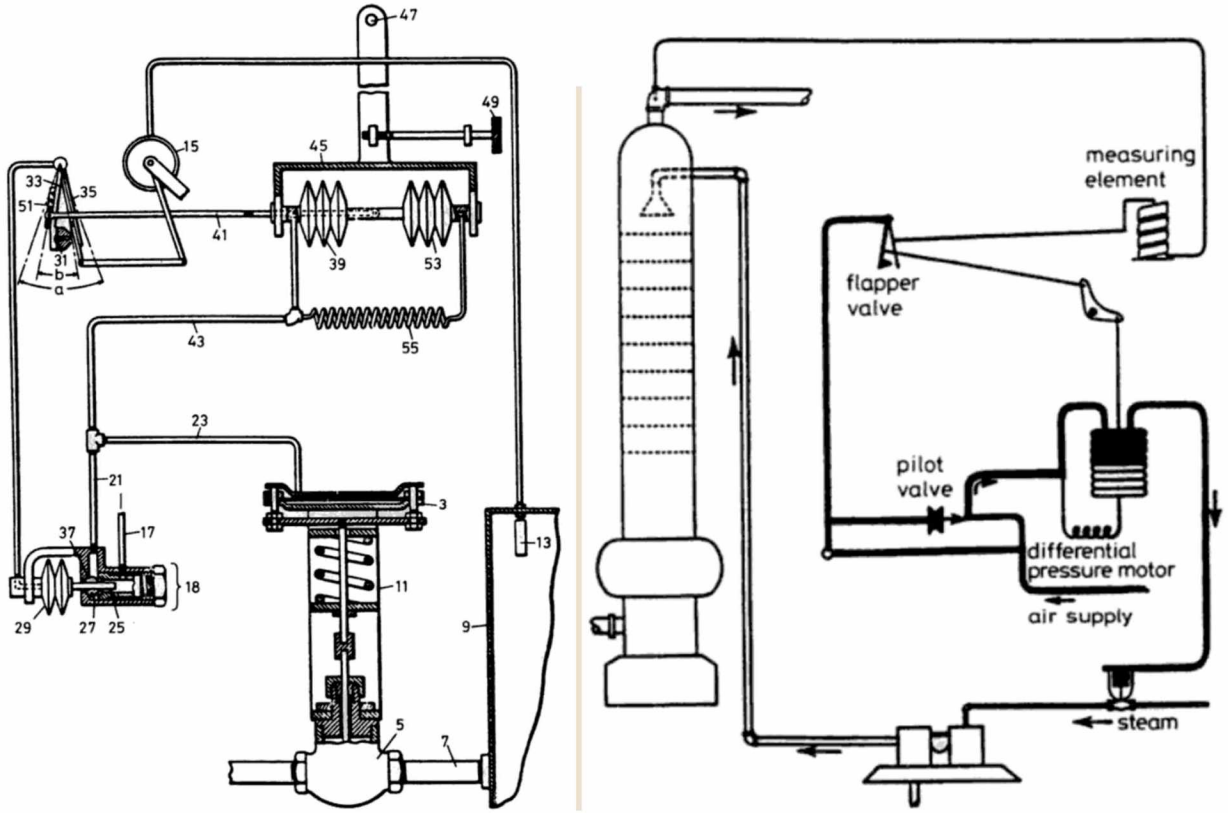


Figure 10. Redrawn Minorsky Controller, circa 1930[37]

2.1.3 Early Autopilot Examples and Theory

Melville Jones, the Francis Mond Aeronautical Engineering Professor of Cambridge University (1919-1952), wrote, "In spite… of the completeness of the experimental and theoretical structure… it is undoubtedly true that… calculations [are] very little used by any but a few research workers. It is in fact rare for anyone actually engaged upon the design of aeroplanes to make direct use of computations." An example of this lack of theoretical understanding is found in Mason's *Stabilog*, Figure 11, which was essentially a pneumatic version of an

electronic negative feedback amplifier.[38] The principle of operation for this device was to control the amount of airflow by how forcefully the air wanted to enter the system. In the common parlance of its time, the output control supplied by the relay valve is regulated, or reset, by diaphragm motor counteracting the control. Specifically, the welded bellows (marked 39 and 53) limited the proportion of the governor valve according to the pressure in the bellows. Thus, the effect is to make the amount an object is instructed to move be proportional to the error and integral of the error. A schematic illustrating this principle of the *Stabilog* is also shown in Figure 11. Mechanical devices such as this, while very powerful for their time, were inevitably limited in their ability to quickly respond to rapidly changing stimuli. In the case of the bellows, time was required to build up the requisite pressure that was needed to properly adjust to the error incurred by its presence.



(a) (b)

Figure 11. (a) Mason's Stabilog Schematic[39] (b) Illustrative Diagram of the Stabilog[40]

Around the same decade as Mason's Stabilog, German companies were developing autopilot systems, represented in Figure 12, in a steady march toward faster and more accurate response times. Askania utilized its experience with early airplane instrumentation (altimeter, magnetic compass, airspeed indicator, etc.) to build a Stabilog style device which had proportional feedback and an air-operated reset, the result of which was not too dissimilar from Sperry's autopilot. A few years later, Siemens, who was hot off of a series of radio-control successes, considered devices that coupled free gyros and clutch-operated electric

servomotors. The turn toward electricity was likely driven by the rapid growth of the industry worldwide, just as embedded computers are a driving force for this thesis. Siemens' final design was an electrohydraulic servo-unit including a rate gyro that offered proportional signal to the angular velocity of the airplane and an electrical moving-coil that acquired heading from a magnetic compass. Finally, building upon lessons learned from the two previous systems, Möller built a complete three-axis controller. Möller even employed clever electro-kinetic techniques such as the proportional gain-constants changing according to a coupled airspeed potentiometer-pickoff; the speed-limits for satisfactory control were thusly increased.[40] One core advantage of developing electrical-based systems, such as Möller's, was servomechanisms. By the end of WWII when anti-aircraft operators were attempting to use radar to detect aircraft, measure range, predict position compared to a shell's velocity, then aim and fire, any measure of rapid control would be highly valued. The capacity to electrically actuate change based upon desired inputs opens doors to systems that can react quickly to a wide variety of inputs.

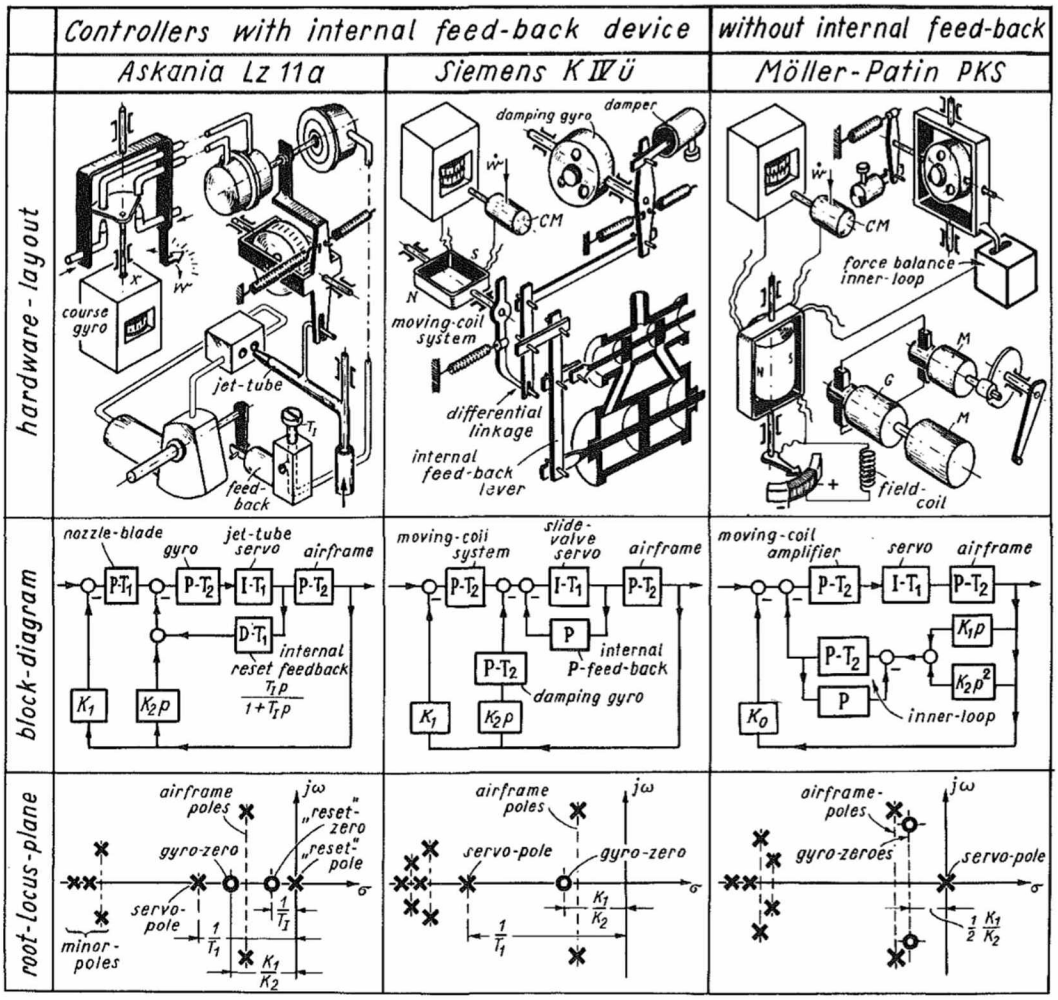


Figure 12. German Flight Control Systems: Pneumatic Askania LZ 11 (left), electro-hydraulic Siemens (center), and electrical Möller-Patin PKS (right)[40]

2.2 Modern and Digital Mechanisms

2.2.1 Modern Autopilot Examples and Theory

Nyquist and Küpfmüller were two early scientists of information in telecommunications, pertinent to the stability problem for a circuit with feedback.[41] First, Küpfmüller offered insights into the dynamics of an electrical feedback system using an integral approach, but was only able to ascertain approximations

and diagrams, when the demand was for rigorous stability. Famously, Nyquist, with the help of his colleague Harold Black, published a frequency-domain encirclement principle that demonstrated, using Fourier transforms, the key to stability was the locus of the open loop frequency response.[42] The significant value of the Nyquist Plots, Figure 13, is the lack of mathematical rigor required to implement this approach; instead a series of measured data points can be used to assess the transient response directly in terms of gains and phase margins. It should be noted that Hendrik Bode took those plots and redrew them linearly along the log-log axes, resulting in even more translatable models for engineers.[43]

The chief initial result of this research was the negative feedback loop itself, fully detailed from the original Bell Labs 1934 report in Figure 14. Harold Black's concept was to deliver the output back to the amplifier in reverse phase—to deter oscillation. His purpose, and the result, was to extend the range of telecommunications signals and thereby greatly improve wartime emergency response electrical signal range, particularly in the transcontinental telephone line. The key to his invention was to cease attempting to stop all distortion on the line, but instead to reduce it at the output of the amplifier.[44] That is, he followed a more robust line of reasoning which is essential to modern autopilots: accept a measure of perturbations but focus on practically controlling them toward the desired goal.

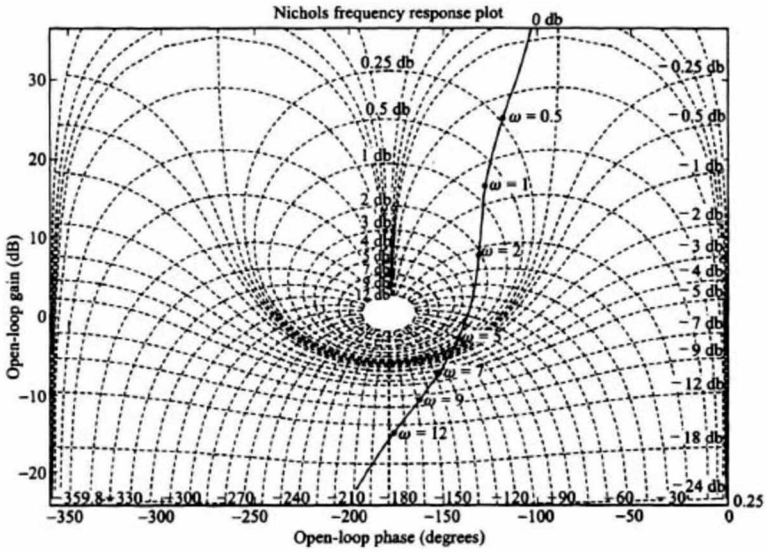
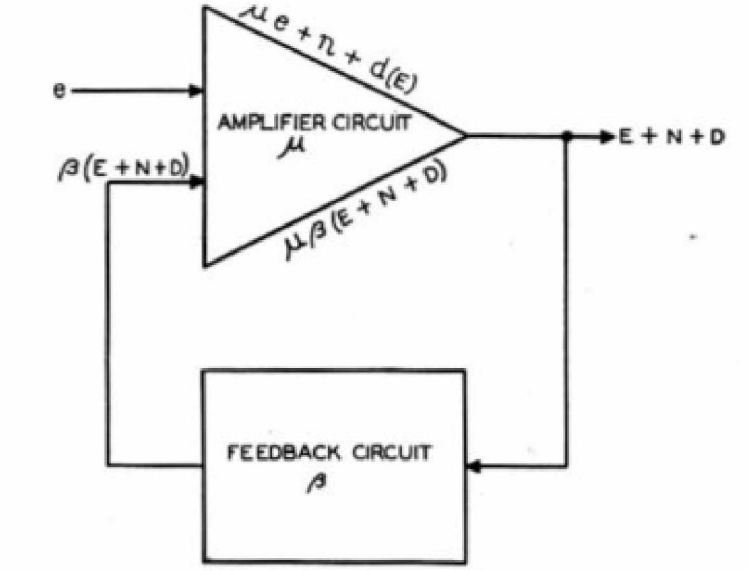


Figure 13. Nichols Chart[45]



e—Signal input voltage.
μ—Propagation of amplifier circuit.
μe—Signal output voltage without feedback.
n—Noise output voltage without feedback.
d(E)—Distortion output voltage without feedback.
β—Propagation of feedback circuit.
E—Signal output voltage with feedback.
N—Noise output voltage with feedback.
D—Distortion output voltage with feedback.

Figure 14. Black's Amplifier system with feedback[46]

2.2.2 State-Space and Programmable Logic Controllers

The continued focus on perturbation control led to the 1950s era matrix models of control dynamics,

$$\dot{\boldsymbol{x}} = \mathrm{A}\boldsymbol{x} + \mathrm{B}\boldsymbol{\mu}$$

$$\boldsymbol{y} = C\boldsymbol{x}, \tag{4}$$

where $x(t)$ is a vector of system states, or internal variables, $\mu(t)$ is a vector of control inputs, and $y(t)$ is a vector of measured outputs. Of course, it is possible to add terms for noise, but this first-order vector differential equation can easily model a multi-input multi-output (MIMO) system as opposed to the more classical theory which was focused on single inputs (e.g. telecommunications). By adding the form,

$$\boldsymbol{\mu} = -K\boldsymbol{x}, \tag{5}$$

it is additionally possible to introduce feedback control with closed-loop properties. Where K is another matrix whose individual elements are control gains in the system. Often, this is called state-variable feedback. Much of these essential early steps, which are suitable for PLCs and microprocessors took place at a series of international conferences and professional bodies of the late 1940s and 50s.[47]

Digital technology in the late 1950s made for tremendous changes to control theory. By 1960, digital computers had begun to process control at a supervisory level, in that the primary loops were still conventional pneumatic, hydraulic, and

electric controllers but were monitored and optimized by a computer.[48] Often these systems were a programmable logic controller (PLC) developed to replace individual relays used for combinational logic in various industries.[49] PLCs couple with state-space driven control theory as they provide adaptive control. For example, gain scheduling, where the gain of a controller is set by some acquired parameter, predates PLCs for flight control. The altitude of an aircraft affects its Reynolds number, particularly with jets or other high-flying aircraft, ergo the gain must change with flight conditions. PLCs and later microprocessors allotted programmed responses to altimeter and pitot tube inputs of a given system, and thereby created modern digital adaptive control mechanisms.

2.2.3 Dubins Curves – Planning Algorithms

A crucial tangential technology to the core digital control theory implemented in this thesis was a paper written primarily from geometrical arguments, by Lester Dubins, which birthed the concept of Dubins Path.[50] Given some constraints such as constant forward speed, maximum steering angle, and minimum turning radius, Dubins' task was to arrive at the shortest path between two points. If the minimum turning radius is zero, then the optimal path in Euclidian space simplifies to

$$L(\tilde{q}, \tilde{u}) = \int_0^{t_F} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2}\, dt, \tag{6}$$

where t_F is the time at which the final point is reached, a given condition is denoted as $\tilde{q} = (x, y, \theta)$, and $\tilde{u}$ is the turn rate control. However, this system can be simplified by constant speed V to the kinematic model,

$$\dot{x} = Vcos(\theta)$$

$$\dot{y} = Vsin(\theta)$$

$$\dot{\theta} = u, \tag{7}$$

where (x, y) is the vehicle's position and θ is the heading. Dubins essentially argues through a series of geodesic geometric proofs and an increasing array of circular paths—building up to four arcs in one of his later proofs, Figure 15—that the shortest path between two points can always be expressed as a combination of no more than three primitives of motion: straight S, left L, right R. As a point of clarity, u is bounded and L and R exist when u is held constant at either bound. He then simplifies the optimal possibilities down to six from paths of length 3:

$$\{LRL, RLR, LSL, LSR, RSL, RSR\}. \tag{8}$$

The shortest path can therefore always be categorized by one of these words, two of which are shown in Figure 16. Dubins also presents a more compressed form of the possible paths by denoting C to mean curve and thusly reduces the effective number of possibilities to two:

$$\{CCC, CSC\}. \tag{9}$$

By making the allowance for the 'straight' path to be zero, further simplification of equations 8 and 9 is possible by removing the curve only paths. Equation 8 reduces to four, and 9 to just one or CSC. However, in practice for this thesis, each of the four words in equation 8 is tested, then the shortest path among them is chosen and followed.

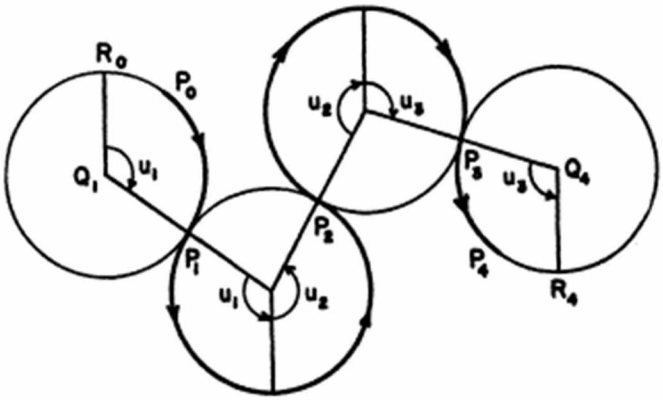


Figure 15. A path of type $CCCC$ used by Lester Dubins to prove his geodesic theory[50]

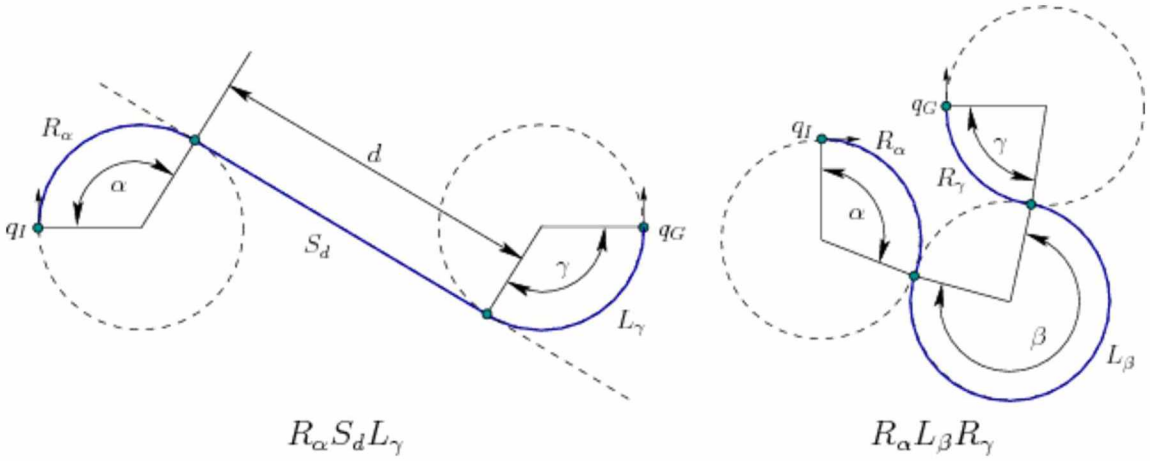


Figure 16. Example of two Dubins Paths: RSL (left) and RLR (right)[51]

2.3 Contemporary Insights

By 2002 computers had advanced and miniaturized the science of PLCs to propel control theory toward the present age. DARPA unveiled a Grand Challenge to test the limits of autonomous control by having vehicles navigate a 150-mile course. A few years later, in 2007, the UAV Outback Challenge was devised to promote and test the viability of unmanned aircraft. A worldwide eruption of research is underway, and as such, the enormity of present relevant research is too vast to be contained herein. Therefore, a fraction of key works imminently preceding and of tangential value to this thesis are described.

2.3.1 Feasibility of Employing a Smartphone as the Payload in a Photogrammetric UAV System[52]

Kim, et al, dive into the possibilities for smartphones as a payload and, by extension, what can be achieved with that system. Smartphones can be operated on the national LTE and 3G networks, and they cost significantly less when compared to a similar payload set of sensors. In a linked study to develop a mapping system besides the LIDAR at 'low cost', the total cost of onboard sensors which the smartphone could theoretically replace—albeit with a drop-in accuracy—was over $25,000.[53] This study collates a suite of feasibility findings into the capacity of a smartphone to correctly acquire images, reduce lens distortion, integrate 3D position, altitude, and orientation data, as well as develop elevation models.

To test the viability of using a smartphone to orchestrate the photographic data collection, a system of three smartphones was used to communicate via SMS and internet protocols to a backend cloud storage and a ground control station, as shown in Figure 17. In all, the system was remotely controlled via SMS then a laptop could monitor, with minimal lag, the results of data gathered and sent to a web server by the commanded aircraft. A limitation of concern in this test was the reliability of the smartphone's GPS, so an auxiliary backup was included to ensure consistent results.

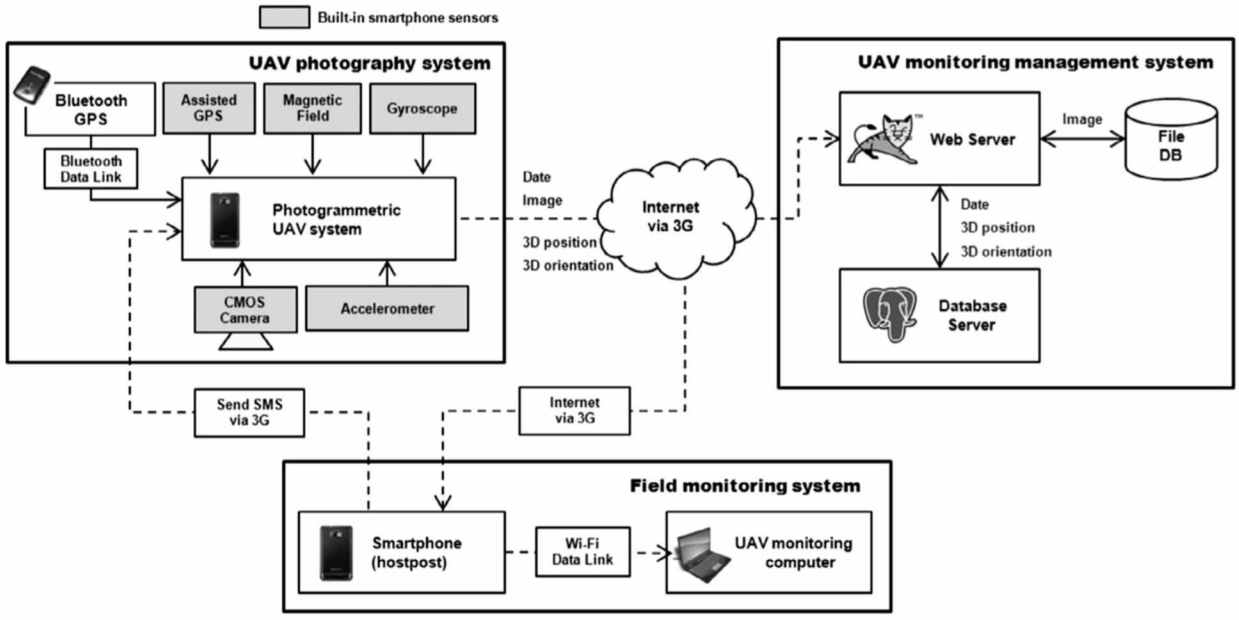


Figure 17. Diagram of data flow for smartphone-payload photogrammetric UAV system[52]

The results confirmed the expectations. Kim, et al, were able to geofence autonomous picture capture by establishing latitude and longitude waypoints within which the smartphone collected images. These images were then overlaid

on Google Maps, within an app on the smartphone itself, as thumbnails. Additionally, they were able to extract macro features such as elevation data from their dynamic tests with a series of images with significant overlap. An example of this data is shown in Figure 18 and hints at the immense accuracy which is possible with just the off-the-shelf systems on a nominal smartphone. They did, however, find that the smartphone's low-cost sensors resulted in accuracy of around 10m with some vibration susceptibility. The capacity to operate even while in telemetrically dark scenarios was additionally lauded. And most of all, they noted that since smartphones can connect with exterior accessories, "autonomous flight using smartphones is expected to be available in the future."[52]

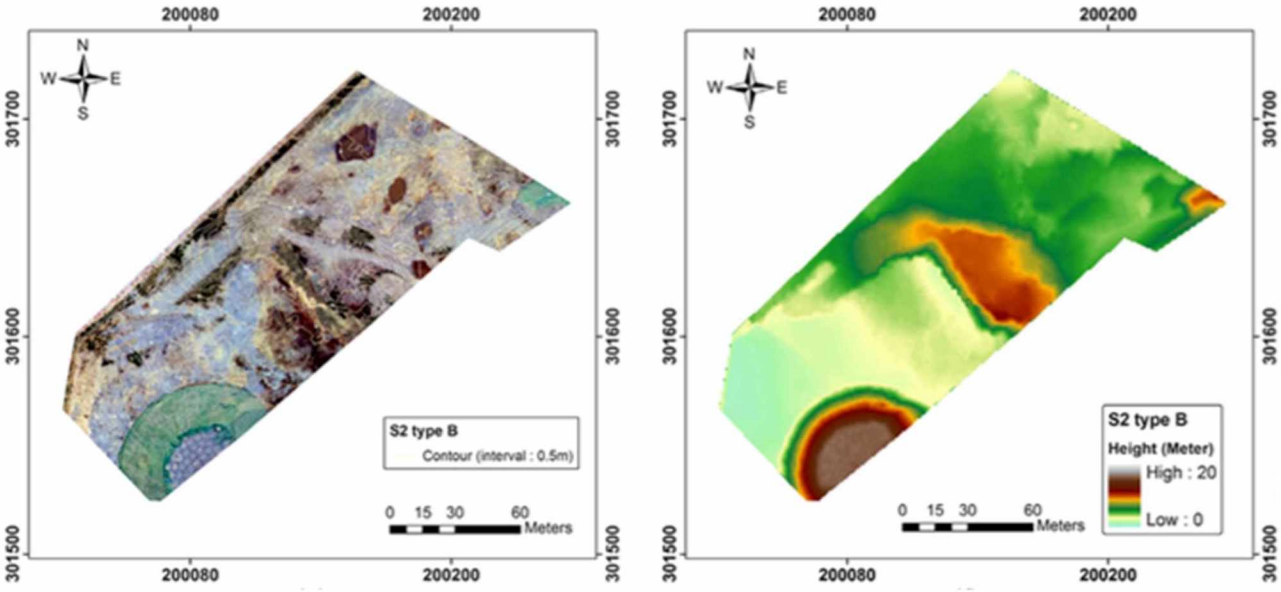


Figure 18. Orthographic and digital elevation model data from onboard Samsung Galaxy[52]

2.3.2 A Blueprint for a Fixed-Wing Autopilot on an Android Smartphone[54]

This paper is primarily an exploration of possibility. The authors Joseph Tabarracci and Patrick Currier note that "the newest frontier in the development of cheaper, unmanned systems, may currently be tucked away in the pockets of millions of people."[54] In order to properly couch the potential of smartphones as the brains of an autopilot, the authors undergo a series of direct comparisons with other off-the-shelf or open-source solutions such as the Paparazzi autopilot.[55] When breaking down the individual sensors between a Nexus and a standard Paparazzi system they find that the tolerances and precision of the GPS, accelerometer, gyroscope, magnetometer, and barometer are within specifications, though they do note that the refresh rate of the GPS is the most significant differentiator and may need augmentation.

To solve the primary limitation of servo command relays, the authors recommend connecting the smartphone to a IOIO (*pronounced "yo-yo"*) which is a Google-employee driven product that is designed to interface external devices seamlessly to the Android marketplace.[56] They also allow for a caveat of airspeed and ground control system (GCS) interfaces which may be a requirement given the concept of operations for the supposed drone. Their potential system architecture is shown in Figure 19.

Though they conclude that this is a highly probable and pursuable stepping stone for the smartphone ecosystem, they take note of challenges: loss of or degradation of GPS, loss of link, and process scheduling. GPS degradation could result in significantly varying intention to mission, and the link disconnect would potentially cause a lost drone. Perhaps most importantly, they note that lack of a real-time operating system on modern smartphones has the potential for unreliable autopilot dynamics. In the end, however, they seem confident that a smartphone autopilot is on the horizon.

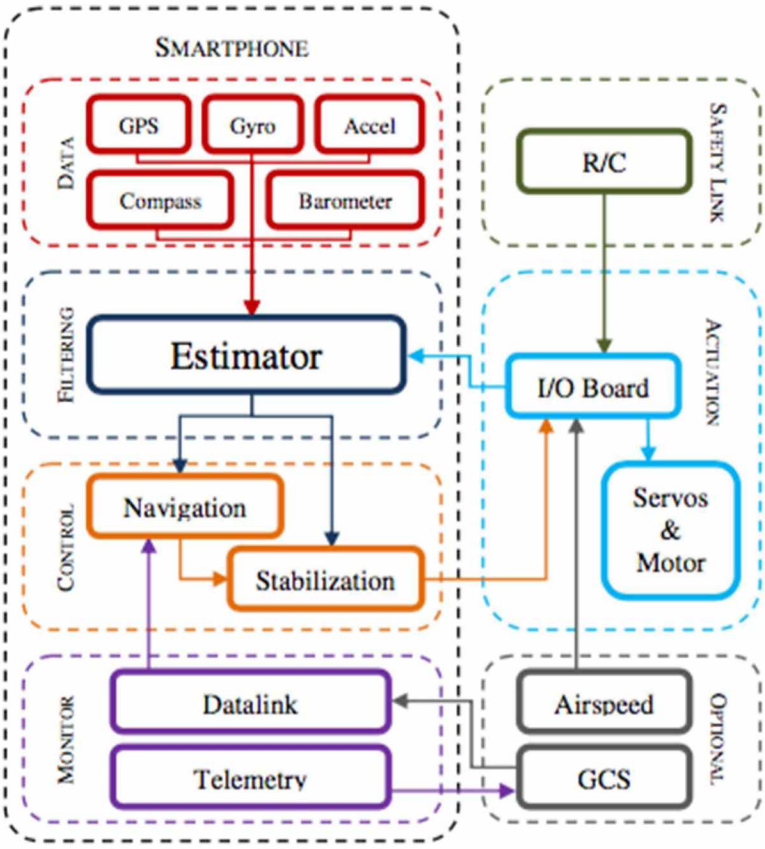


Figure 19. Tabarracci proposed smartphone autopilot system architecture[54]

2.3.3 Stabilization and Control of Quadrotor Helicopter Using a Smartphone Device[57]

Desai, Lee, et al, at Brigham Young University propose utilizing a smartphone as the sole device for a quadcopter autopilot. Specifically, the authors look to optimize the immense computational potential to not only command the necessary servos but also cogitate vision detection algorithms to determine their position in real time. The authors use pulse-width modulation (PWM) signals via a IOIO to command the servomotors alongside vision algorithms such as Harris feature detector and RANSAC similarity-constrained homography.

By color-coding the edges of the quadcopter, they created a vision detection condition whereby the smartphone could accurately determine its orientation in 3D space. However, due to the realities of applied vision systems, introduced noise can often derail the system. To combat this and improve the robustness of the platform an Extended Kalman Filter (EKF) was implemented. An EKF essentially linearizes using a stochastic estimate of the data from a non-linear reality.[58] This was done in a loop in concert with a derivation outlier reject, and the result was the model's ability to properly estimate the center of mass.

The onboard processing proved successful, and the authors were able to stabilize a quadcopter using the vision and computing capacity of a smartphone. Someday, they would hope to extend these essential features beyond the laboratory and into outdoor environments.

Chapter 3: Technical Details and UGV

Before the project could fly, it had to drive. The first stage toward developing an unmanned ground vehicle (UGV) is building and verifying the fusion algorithm. The fusion algorithm is a collective term referring to the methodology that collects and integrates the data from sensors, then combines that information into a context that can be utilized to establish a predictive framework of velocity and position. The core aspect of the fusion algorithm for this thesis was based upon a proportional-integral-derivative (PID) controller.

3.1 Typical PID

In a macro sense, there are two main types of control systems: open loop and closed loop. The open system is dependent directly on the inputs. As a simple example, an operator can turn a variable resistor to alter the load on a motor which will change its speed. However, the operator has no knowledge of the speed change induced in the motor by the new loads, so therefore no feedback is present to influence the outcome. Correspondingly, a closed loop system introduces a cycle of information whereby the operator is informed of the changes induced by the

new loads and can adjust accordingly. This feedback loop can exist in a diversity of forms and a PID falls within this category.

Figure 20 shows a block diagram of the generic PID control system. A PID attempts to the eliminate the difference between the desired value of a parameter and its measured value. The three mechanisms of the PID controller are described below:

- *proportional*—This term is a linear response based on the parameter error, centered at zero.
- *integral*—Often the system will require a non-zero response to maintain equilibrium. This term is an accumulation of the proportional error over time, which will remain present even after the error is eliminated.
- *derivative*—This system is similar to the proportional term, but instead provides a linear response to the time-derivative of the parameter error, centered at zero.

All three of these terms are not necessarily used for each controller. Some controllers utilized only the proportional and integral terms (PI) or only the proportional and derivative terms (PD). For example, command roll angle using ailerons was a full PID, but command pitch angle using elevator was only PD and the command airspeed using throttle was only PI.

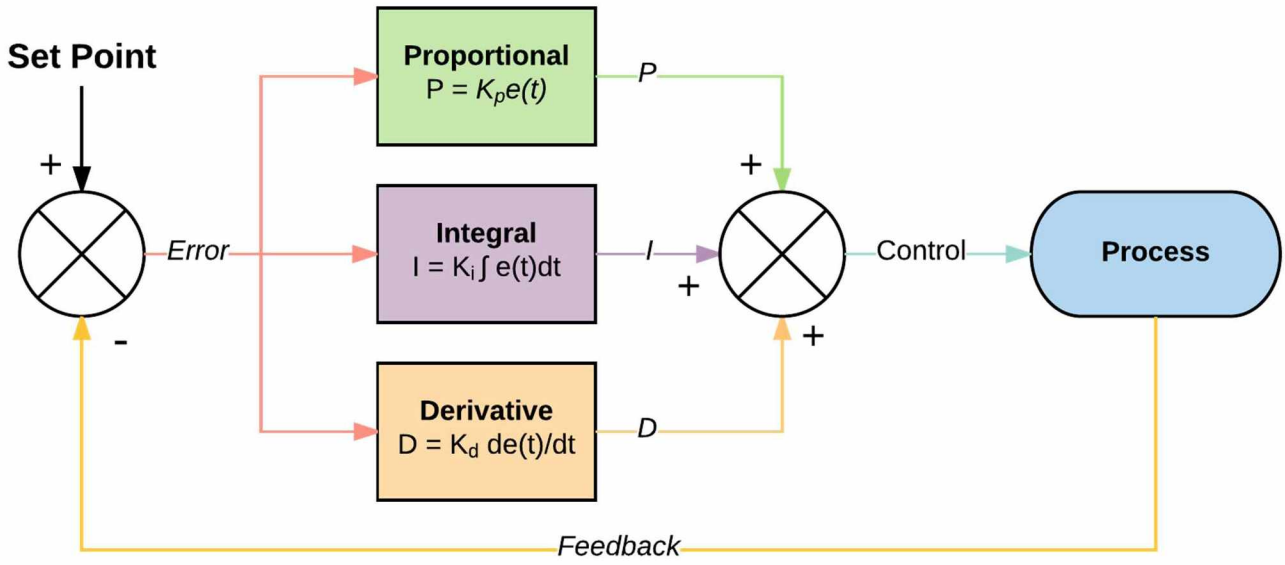


Figure 20. PID block diagram

In general, larger magnitudes of the constant terms (often referred to as gains) would lead to faster system responses, but also larger overshoots and settling times. Rather than attempt to predict the appropriate gains for each controller using a system model, the gains were developed through extensive vehicle testing. The essential strategy was to increase each gain one at a time until the vehicle began to exhibit undesirable traits. By progressing through each controller separately, it was possible to identify the effects of altering a single gain. This proved to be more effective than the system modeling approach, as the models created were never accurate enough to sufficiently predict the controller gains. However, future work could utilize adaptive control techniques to eliminate or reduce the large testing phase required to identify ideal PID parameters.

3.2 Primary Fusion Logic

Once the vehicle had been launched, there was a single loop which controlled all autopilot functions, each of which occurred at a specified interval. The pseudocode for this loop is shown in Figure 21; based upon work done previously by Premerlani and Bizard.[59]

The vehicle attitude, position, and velocity were computed during every loop. The gyroscope measurements were used to calculate rotational movement, while the accelerometer measurements were used in conjunction with a fusion algorithm to account for gyroscope drift. With the updated direction cosine matrix,

$$\boldsymbol{A} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi + \sin\varphi\sin\theta\cos\psi & \sin\varphi\sin\psi - \cos\varphi\sin\theta\cos\psi \\ -\cos\theta\sin\psi & \cos\varphi\cos\psi - \sin\varphi\sin\theta\sin\psi & \sin\varphi\cos\psi + \cos\varphi\sin\theta\sin\psi \\ \sin\theta & -\sin\varphi\cos\theta & \cos\varphi\cos\theta \end{bmatrix} \quad (10)$$

the earth-frame velocity and position could then be estimated.

With every new GPS measurement, the differences between the predicted and measured position/velocity were used to create corrections that were incrementally applied until the next GPS measurement was available. In this way, the GPS data was treated as the truth source.

Because GPS altitude was deemed to be too inaccurate, pressure altitude was used instead. This was obtained by means of a barometric pressure sensor and applied directly to the vehicle position. To estimate the vehicle's height above

ground level (AGL), the pressure altitude at takeoff was set as a reference point, thus any altitude above that was above ground level.

```
while (flying)
        update vehicle attitude
                read 3-axis accelerometer
                read 3-axis gyroscope
                update direction cosine matrix (DCM)
        update vehicle position and velocity
        if (new GPS data)
                read GPS position and velocity
                update vehicle position and velocity correction
        if (new barometric altitude data)
                read altitude sensor
                update vehicle altitude
        check for waypoint completion
                if (current waypoint reached)
                        update current waypoint
        compute vehicle control commands
                calculate desired path based on current point
                calculate deviation from desired path based on current attitude/position/velocity
                calculate control inputs required to eliminate deviation
        move vehicle actuators
        transmit current vehicle status to ground station
        check for message from ground station
                if (new message)
                        perform required action
end while
```

Figure 21. Fusion algorithm pseudocode

The vehicle's location was frequently compared to that of its current destination waypoint to determine when the waypoint had been achieved. The vehicle was not just trying to head towards the waypoint, but was rather attempting to travel along a three-dimensional path. Depending on its current position and

orientation, a desired course and altitude were calculated. These were then translated into actuator commands which were passed on to the actuators. The vehicle would send status messages which included basic attitude, position, velocity, and path information to the ground station. The ground station was also able to send commands to the vehicle, such as "start mission" or "return to base".

3.3 Then it Drove

All primary functions of the autopilot (estimations regarding altitude, position, and velocity as well as waypoint navigation and actuator control) were first tested on a ground vehicle before being deployed in an aircraft. The UGV, see Figure 22, provided a few distinct advantages for the sake of testing, the most notable being that in the event of a program malfunction, the throttle could be cut and the vehicle would simply come to a stop. This was, of course, in contrast to the air vehicle which would have to be manually recovered in flight, a feat that was not always possible. The other significant benefit of the ground vehicle was that it was always pointed approximately in the direction in which it was traveling. By aligning the vehicle course (direction of travel) and heading (direction it was pointed), it was much easier to diagnose problems. In this way, any discrepancy with the heading could automatically be attributed to a discrepancy with the course, which was being estimated using GPS data.

Figure 22. Unmanned ground vehicle with smartphone controller

Besides the inherent differences in vehicle dynamics, the path-following logic was the same for both the ground and air vehicles, with the exception that the aircraft was required to control its altitude, while the ground vehicle remained at zero feet AGL regardless of its inputs. The path-following tests were also greatly aided by using a ground vehicle, as it was possible to always record the precise position of the vehicle. This could be compared to both the predicted position as well as the desired position to determine if the autopilot was able to know where it was, and know how to get where it wanted to go. Path-following was much simpler with the ground vehicle, as control was accomplished by means of a single steering mechanism, thus the algorithm logic could be analyzed more completely without the concern that vehicle dynamics might be playing a significant role in the results.

Chapter 4: Experimental Design

With the fusion algorithm established, the primary experiment of this thesis was ready for testing. This chapter outlines the construction of the experimental UAS controlled by an onboard smartphone.

4.1 Setup

In order for the autopilot to physically control the aircraft, it was necessary to find a device that could interface between the smartphone and the hardware, to relay the motor and servos commands, as well as connect the smartphone to any auxiliary components such as the barometric pressure sensor. The IOIO board, designed specifically for Android, was chosen for this purpose.[60]

As shown in Figure 23, the hardware used was a Nexus 5 smartphone connected via USB to the IOIO board that interfaced with the servos (five channel: elevator, ailerons, rudder, and speed controller), barometer, and XBee transceiver. The XBee transceiver is used as the communication platform for telemetry and for receiving commands from an Android tablet running the same application in ground state mode. It had a one-mile range line-of-sight (LOS) conditions and was

additionally used during early experimental phases to keep a human in the loop as a failsafe against an impending crash. The external camera was a FlyCamOne eco V2 digital camera, capable of 480p resolution. Although the smartphone has a camera with superior quality onboard, the external camera allowed for mobility in lens location to simplify testing as a smartphone's image capture capacity is not in doubt. Similarly, the decision was made to connect an external barometer since most smartphones do not have a barometer installed; though the Nexus 5 does, the Eagle Tree Systems barometer had a superior resolution of 0.1 meters.

The physical aircraft, Figure 24, used to test the autopilot was a commercial-off-the-shelf (COTS) Parkzone T-28 Trojan; modified as required. The T-28 had a physical wingspan of 44 inches with a flying weight of approximately 42 ounces. It utilized an electric brushless motor powered by 2200 milliamp hour lithium-polymer battery. All of the hardware components, including the smartphone, fit inside the custom cavity of the fuselage. Every sensor employed, including those internal to the smartphone, was within tolerance of COTS autopilots.[54]

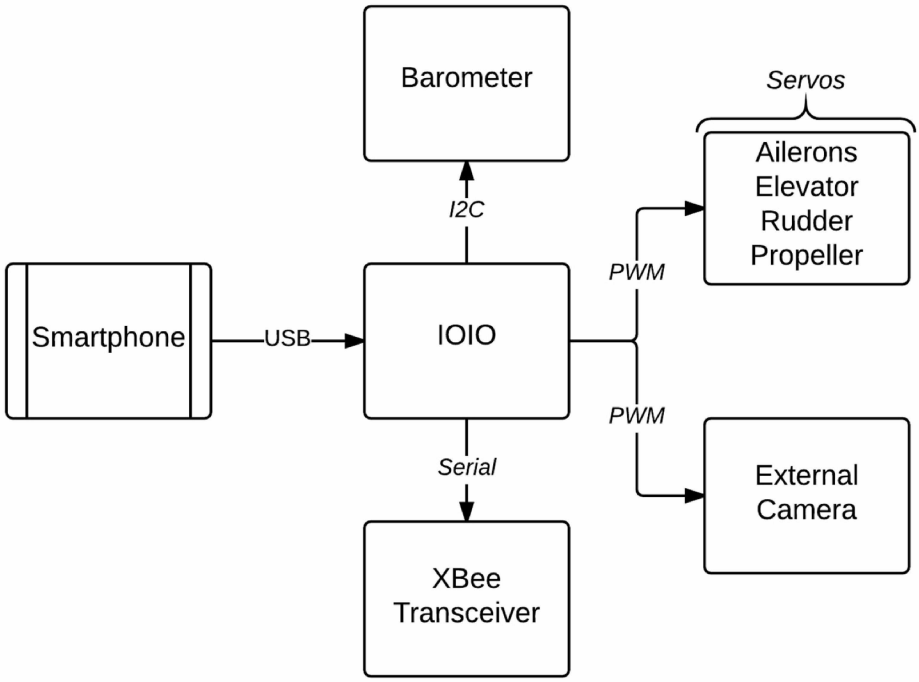


Figure 23. Overview of UAS hardware

Figure 24. UAS experimental platform (Parkzone T-28 RC airplane)

4.2 The Control Software

Figure 25 delineates the finite states of the autopilot application for a nominal mission. The UAS would take flight, climb to mission altitude, reach the provided waypoints, descend to landing altitude, then safely approach the runway, and land. Note that no takeoff or landing assistance was necessary—e.g. catapult, hand-launch, retrieval net—as the aircraft was able to utilize its landing gear for these phases. Two interrupt commands were also provided to the ground control tablet: loiter and land. These allowed the user to alter the mission profile.

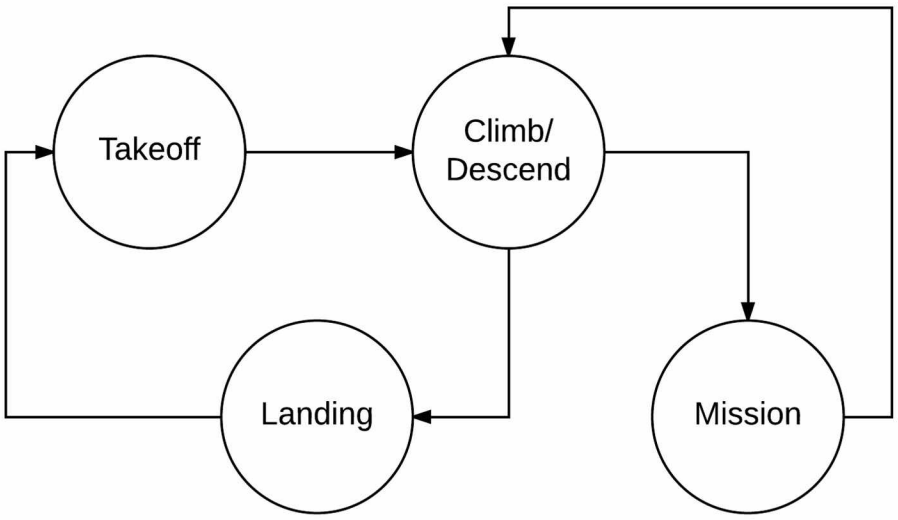


Figure 25. UAS finite state diagram

Each of these finite states required four types of information to create the GPS-aided Inertial Navigation System (GPS/INS): acceleration, rotation, direction of travel, and barometric pressure. All of these pieces of information could be obtained using the hardware installed on many smartphones; however, as

aforementioned, barometric pressure is not ubiquitous, so an external pressure sensor was utilized. The adjustments for flight were determined by these four sensors providing the key inputs to the inertial navigation system, as shown in Figure 26. The data is looped and filtered through the waypoint navigation whereby commands are sent to the servos. Two-way communication with the ground station relays aircraft status while allowing new missions directives to be initiated.

As initially outlined in Chapter 3.2, the attitude of the aircraft was calculated by combining the data taken from a 3-axis gyroscope and a 3-axis accelerometer, each of which was sampled at approximately 200 Hz. The gyroscope readings were used to update a direction cosine matrix, from what roll, pitch, and yaw could be obtained. The accelerometer was used to account for any gyroscopic drift, as well as to estimate the aircraft position and velocity in between GPS updates. Latitude, longitude, and height above WGS84 reference ellipsoid were available from the GPS receiver at 1 Hz. The pressure altitude was calculated using barometric pressure sensor which was sampled at approximately 20 Hz. Combining the data from these sensors, the altitude and latitude/longitude of the aircraft were calculated at approximately 200 Hz, while the absolute height was calculated at nearly 20 Hz. Corrections to the calculated location were applied based on the most recent GPS data received.

It should be noted that a magnetometer was not used to determine the aircraft heading in order to avoid recalibration before every flight and to remove the potential for magnetic interference that could result in significant navigation errors. Instead, it was assumed that the aircraft was always heading in approximately the same direction that it was traveling, ergo the heading and the course (direction of travel) were forced to converge, with the course being the dominant component. A safety R/C override was included during testing which could be activated by a transceiver in case of loss of control.

One of the primary challenges with using a smartphone is that the autopilot program must run on a non-real-time operation system. Typical autopilots consist of a real-time operating systems (RTOS) designed to actively monitor flight.[61] An RTOS has a high degree of consistency in terms of the time it takes to perform a specific task. This is of particular importance when sensor data is required at reliable intervals. Unfortunately, with a non-RTOS, depending on the processing resources available at any given moment, computation time can vary, which can negatively affect the performance of the autopilot. However, it was found that the autopilot was capable of controlling the aircraft despite minor variances in processing performance as the sensors were measured at a consistent rate. This was due to two factors: the high-performance capability of the current smartphones and the inherent stability of fixed-wing aircraft.

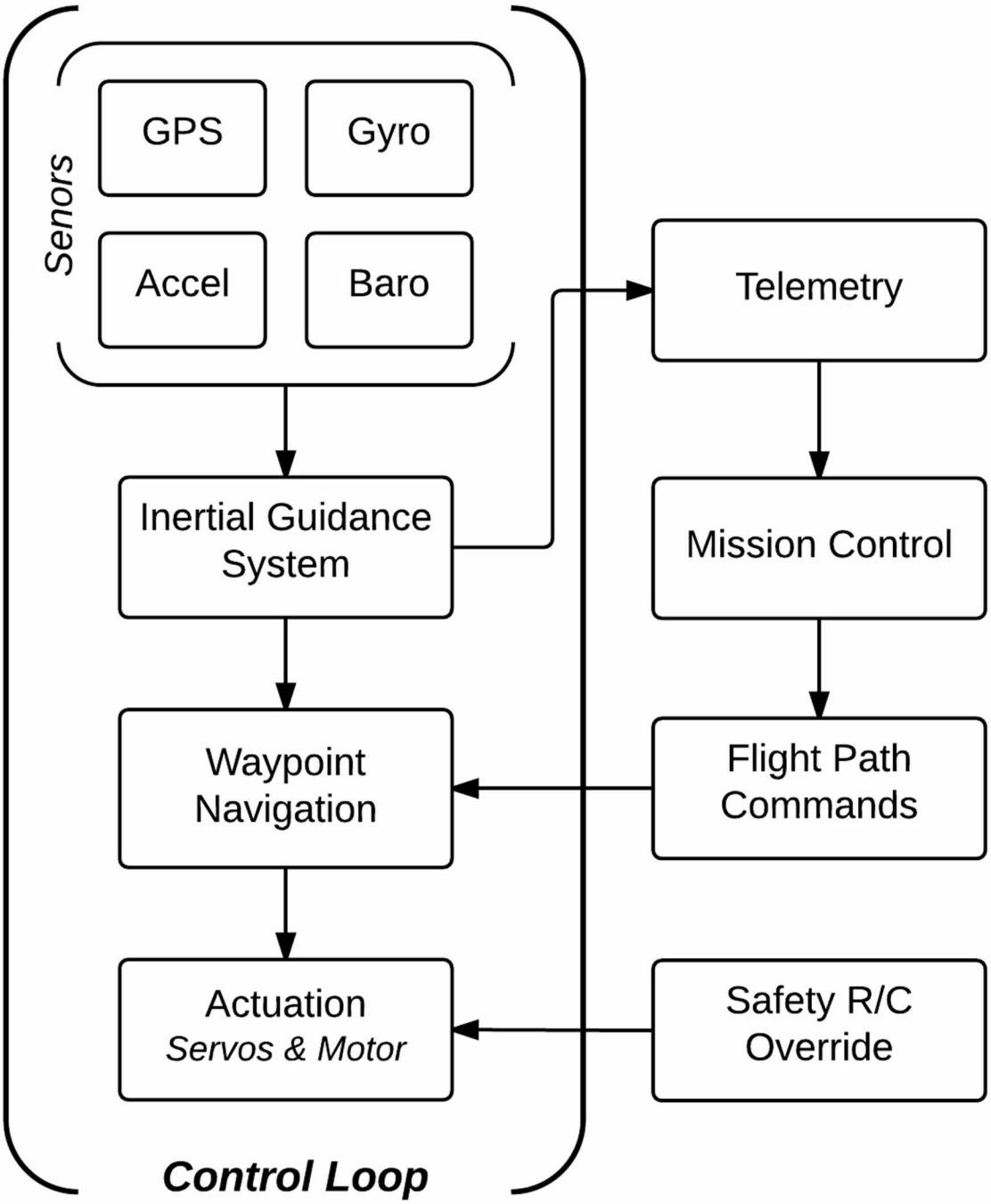


Figure 26. UAS software diagram

The Nexus 5 smartphone used for the autopilot contained a Qualcomm Snapdragon quad-core 2.26 GHz processor and 2GB of RAM. This hardware was capable of rapidly completing intense computational tasks, thus the possibility of one task delaying another was negligible. The main cause of timing inconsistency was due to sensor data readings. Although the sensors were polled at consistent intervals by the operating system, the data was occasionally delayed. In other words, the time between measurements did not match the time between measurement availability. While this could be a problem with a highly unstable system, the dynamics of the test aircraft were such that small delays in control inputs could be tolerated. Nominally, if the aircraft were well trimmed, it would tend to continue flying in a similar manner unless control inputs were applied. These small inconsistencies in operating system timings are discussed further in Chapter 5.3.

Chapter 5: Results

This chapter describes the key findings from the construction of the UAS.

5.1 Application and UI

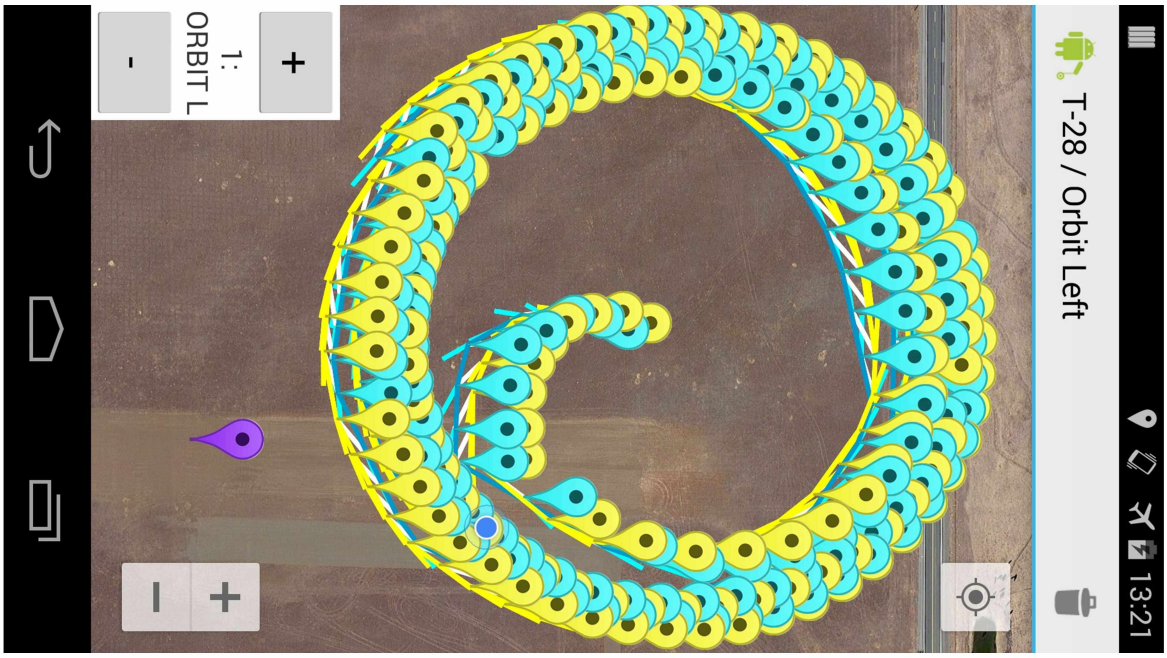
Examples of the graphical user interface (GUI) of the UAS's Android autopilot app, written in Java, are shown in Figure 27. The GUI depicted in Figure 27 (a) displayed autopilot attitude, position, current flight mode, and other essential parameters, allowing for immediate detection of any abnormalities. The interface also allowed for a mission to be initiated without the need for a ground station, since the phone was placed in an accessible location of the UAS. Figure 27 (b) demonstrates how declaring a series of waypoints created missions. Waypoint coordinates were entered relative to a single latitude/longitude position set by the user. Flight results were available upon mission completion, as shown in Figure 28 (a) and (b). Yellow pins represented the location used by the autopilot, while blue pins were GPS measurements. The lines originating from each point show the respective course over ground. The purple pin is the initial takeoff location. As the figure delineates, the aircraft position as recorded by the autopilot GPS compared to the intended mission established path following capability.

7:44
0.0
350.2
0.0
0.0
0.0
A/S
Alt
Wind N
Wind E
Wind D
42.31700
-122.96162
448
0.0
-57.3
Latitude
Longitude
h^GPS
Vg
Course
0.0
448
14.0
0.40
Takeoff Mode
Vg
Height
Hdop
Latency
-180
1
195
195.5
Roll
Pitch
Yaw
Course
-1.2
0.5
-0.1
p
q
r
Stop Navigation
Reset Navigation
Launch Mission
Manual Flight
Manual Flight
Flip: 3
GPS
1,1
2000
1.00
10000
0.50
φθ^kp
φθ^ki
ψ^kp
ψ^ki

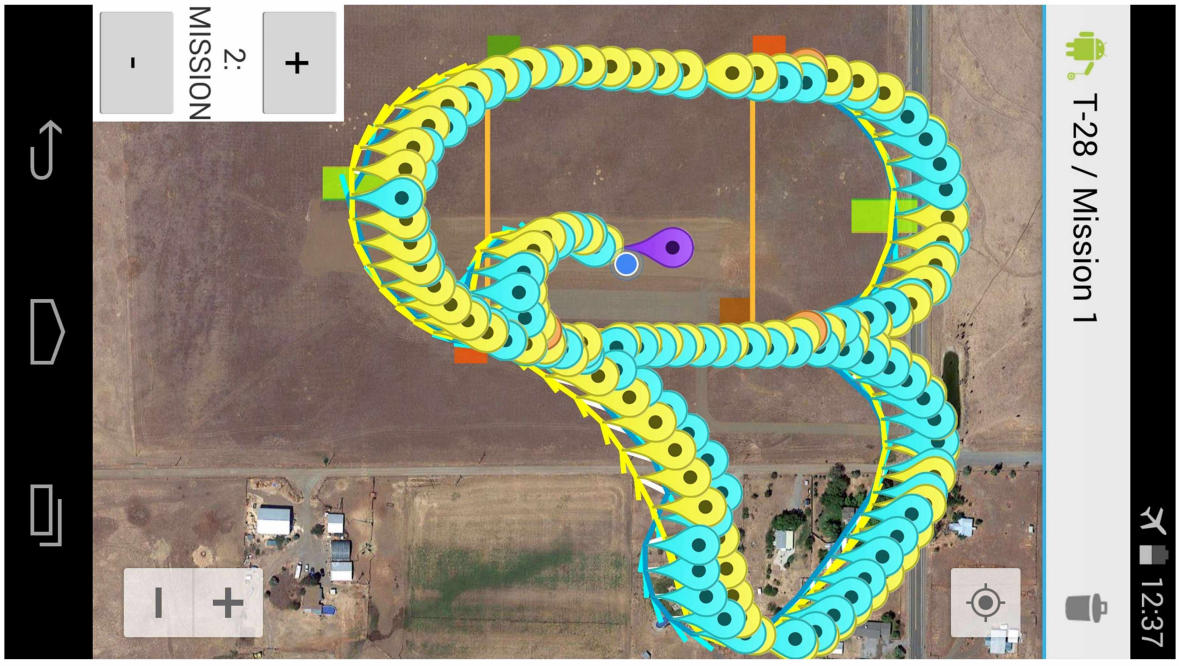
(a)

(b)

Figure 27. Examples of UAS (a) interface and (b) mission waypoints



(a)



(b)

Figure 28. UAS flight results for (a) orbit and (b) full four waypoint mission

5.2 Flight Performance

The aircraft successfully completed many autonomous flights. These flights ranged from short duration commanded altitude to full missions with several waypoints. The autopilot was able to complete out-and-back missions of approximately ten miles, with a duration exceeding fifteen minutes. It could be commanded to take images and would autonomously generate its own Dubins-derived flightpaths (see Chapter 2.2.3) based upon the waypoints and the turns required to meet them.

Key tests of the autopilot's robust design were completed. For example, in order to demonstrate the ability to maintain control after a modest perturbation, the UAS was successfully commanded to transition from wings-level upright flight to wings-level inverted flight by means of a 180° aileron roll. More generally, entire missions were captured by the autopilot-commanded undercarriage camera, some video stills from which are shown in Figure 29. Ground footage was shot manually, shown in Figure 30. A compilation video of several successful flights can be found online.[62]

Since absolute truth can more easily be measured on the ground, the UGV discussed in Chapter 3 was tested. The car drove along a straight line continuously with a cross-track error of less than 2 meters. Image stills from one such successful test are shown in Figure 31, and a full video can be found online.[63]

Figure 29. Video still from undercarriage camera captured by UAS[62]

Figure 30. Video stills from manually shot ground footage[62]

Figure 31. Video stills from UGV path following tests[63]

As aforementioned, the low-resolution camera equipped to the UAS could be actuated by the autopilot. Successful capture of these images required relatively stable in-flight conditions. Figure 32 depicts the measured versus commanded altitude of an entire mission from takeoff to landing. The UAS buoyed and sunk due to aerodynamic forces, but stochastically followed its intended altitude.

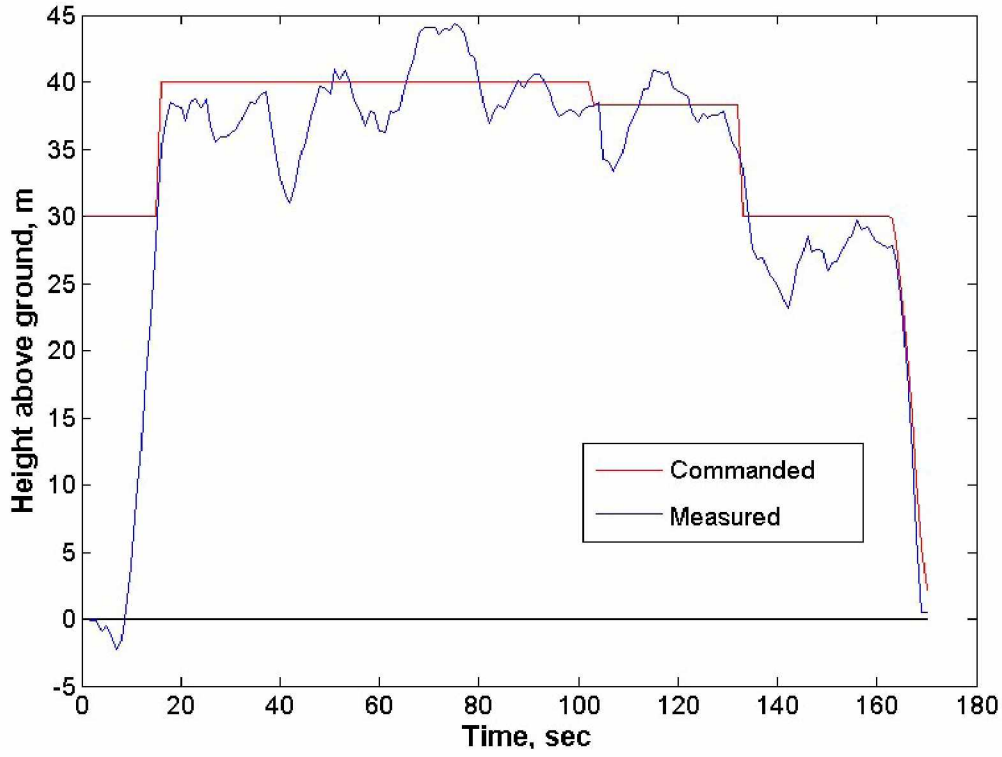


Figure 32. Commanded versus measured elevation

Some of the altitude variance is due to the natural inconsistency of barometric pressure, and this uncertainty is most evident in landing accuracy. Figure 33 shows landing touchdown positions compared to intended, as collected from a typical day of testing. This particular test error was found to average 22m

radii from the commanded location; the error tended to be more longitudinally than laterally, which is to be expected of imprecise landing accuracy.

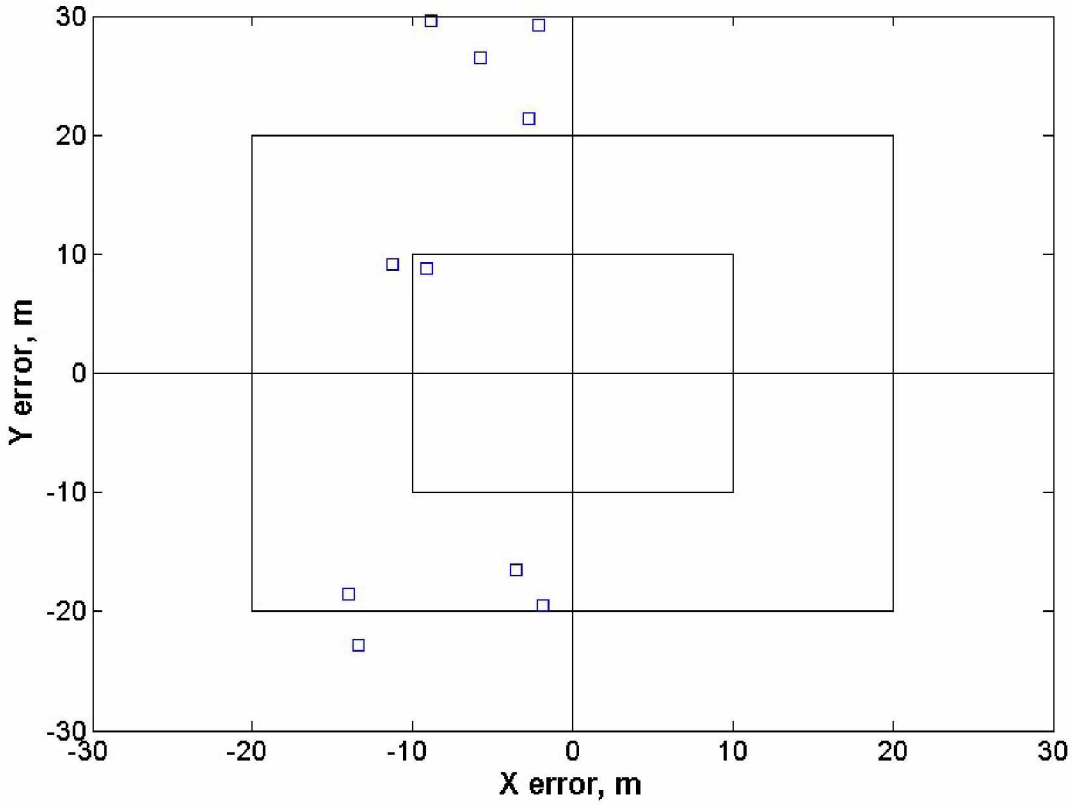


Figure 33. Scatterplot of UAS landing accuracy

5.3 Non-RTOS Analysis

Because the Android smartphone did not utilize a RTOS, it was necessary to check the consistency of the OS timing as the autopilot needed its functions to be called at reliable intervals. Therefore, the autopilot was monitored for several hours of continuous operation. The primary navigation loop was commanded at intervals of 5 milliseconds, and the servos were commanded at intervals of 50 milliseconds. The mean and median values were exactly as instructed. Although the interval

length did fluctuate, any lead or lag from one was corrected by the proceeding so that the average of 200 navigation loops or 20 servo loops was still 1 second. Furthermore, the largest measured intervals were within the bounds of acceptable delay for a stable fixed-wing aircraft with the navigation loops and servo loops experiencing maximum intervals of 30 and 87 milliseconds, respectively.

The operations that occurred most frequently were accessing the accelerometer and gyroscope data, which was supposed to happen at 200 Hz. Small variations in the timing were considered acceptable, but it was important that any lag not be allowed to propagate and instead be eliminated in subsequent iterations. Figure 34 depicts the system lead/lag caused by accessing the accelerometer and gyroscope sensors over the course of 100 seconds, measured in terms of percentage of the desired interval, which was 5 milliseconds. In other words, a 5% lag would mean that the loop took 0.00025 seconds longer than it should have. The figure clearly shows that any lag was quickly countered by a lead, so that the overall timing of the system was quite consistent, with the absolute error remaining very small. Timing inconsistencies were accounted for by using the timestamp associated with each sensor measurement.

The other important repeating function was the servo actuation which was intended to operate at 20 Hz. The requirements for this timing was less strict, as 20 Hz is much faster than a human would be applying control inputs, so even

relatively large lags (assuming they didn't accumulate) could be tolerated. Figure 35 demonstrates that the maximum lead/lag between servo actuations is approximately 0.010 seconds, which was considered to be acceptable.

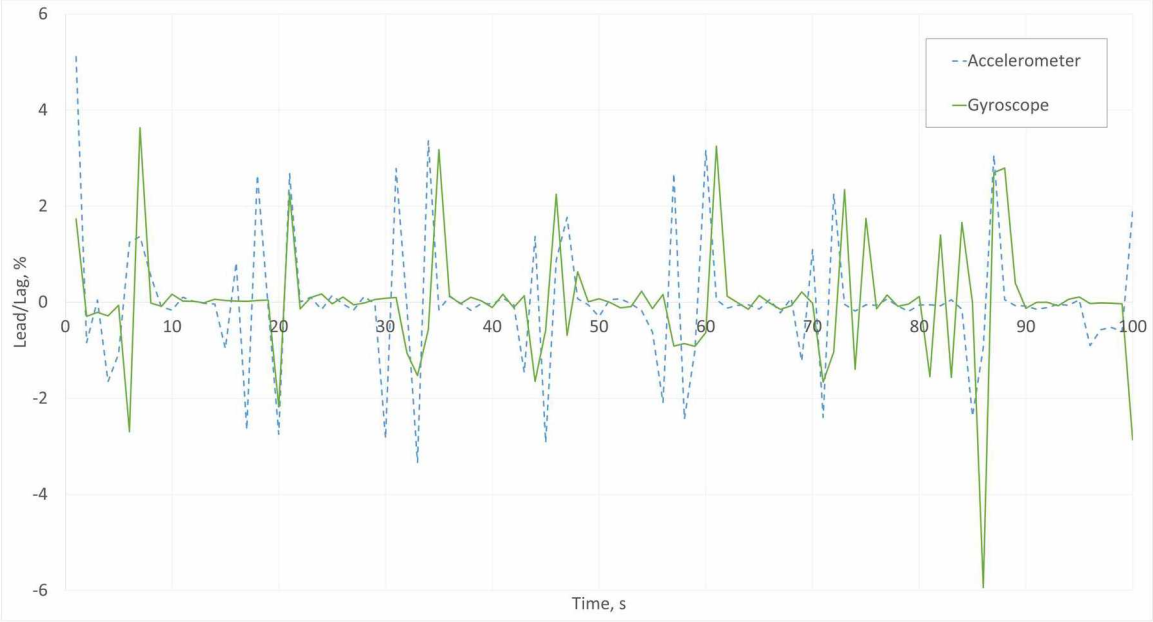


Figure 34. Accelerometer and gyroscope timing variation

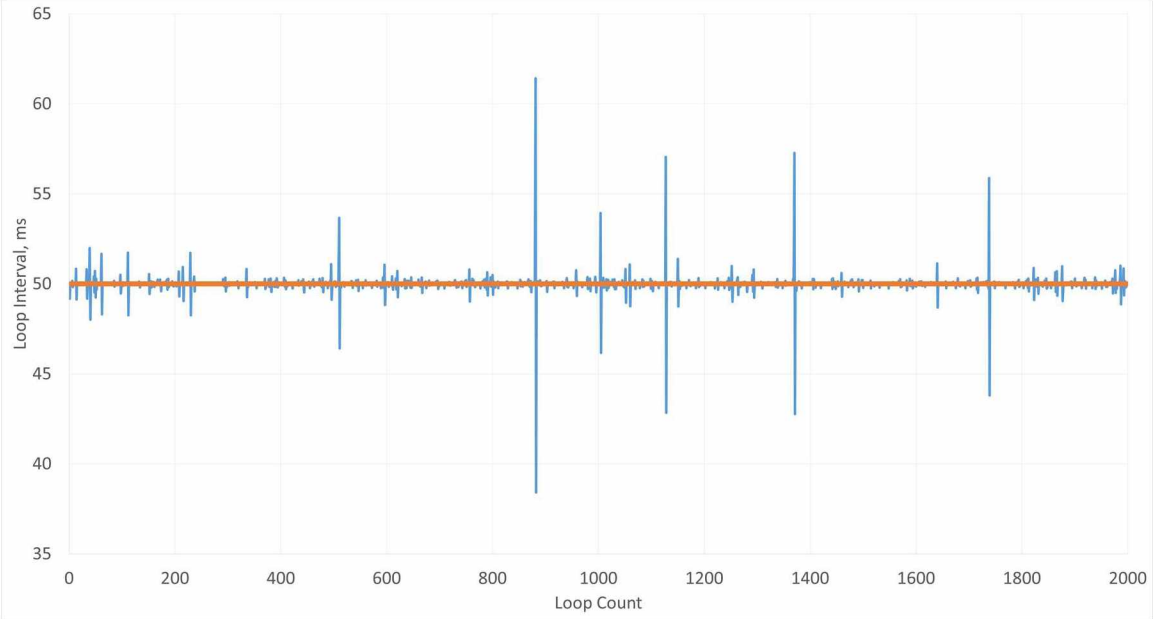


Figure 35. Servo timing variation

Chapter 6: Additional Explorations

During the developing of this thesis, a lot of extraneous investigations were conducted. This chapter is devoted to a few of the intriguing corollary outputs from those inquiries.

6.1 Unscented Kalman Filter[64]

Aircraft heading was one of the more difficult parameters to estimate. While magnetometers could potentially provide the vehicle's magnetic heading, they often suffer from magnetic interference and require frequent calibration. These undesirable characteristics led to the exploration of other methods to estimate heading.

To obtain a rudimentary heading estimation with a fixed-wing aircraft, it could be assumed that the aircraft was generally pointed in the direction it was travelling. This was found to be satisfactory in calm conditions but was unreliable when sufficient wind was present. A more thorough method utilized an Unscented Kalman Filter (UKF) which coalesced GPS, inertial, and airspeed measurements to estimate the aircraft heading as well as wind velocity.

The fundamental equation utilized by a UKF was derived from the "Wind Triangle", shown below, which states that the vehicle's ground velocity is equal to the sum of its airspeed vector and the wind vector.[64] The airspeed could be measured by means of a differential pressure sensor using the vehicle's attitude, while the ground velocity was estimated using data GPS. Of course, the vehicle's heading was required to convert the measured airspeed from a body-frame measurement to an Earth-frame vector, and thus the equation could not be solved explicitly. The UKF was used to overcome this problem.

$$\vec{V}_a + \vec{V}_w = \vec{V}_g \tag{11}$$

This method was never tested using actual airspeed measurements but rather airspeed estimations. Although this produced decent results for the heading estimation, it was decided that it was too inaccurate for use as the primary heading estimation device.

6.2 LIDAR[65]

To accurately determine the height above ground level (AGL) during landing, a LIDAR sensor was used (Figure 36). The word LIDAR is a portmanteau of light and radar, since it uses the principle of a radar but instead uses light from a laser to determine distance. The PulsedLight LIDAR was capable of 2.5cm accuracy at distances up to 40m and proved very proficient at providing AGL data,

which was accessed at 10Hz. The sensor communicated by means of the I2C protocol, the same as the barometer; ergo, it was simple to connect to the IOIO.

Figure 36. PulsedLight LIDAR for improved landing accuracy[65]

Before using LIDAR, landings were performed by setting the aircraft on a constant-slope descent until it finally reached the ground. This method meant that either the landing distance was very long (shallow descent) or the impact speed was very high (steep descent), both of which were undesirable. With a reliable AGL source, the aircraft could make a steep descent until getting very close to the ground, at which point it could descend much more slowly, resulting in short landing distances and a gentle impact.

6.3 Path Following

One limiting factor for seamless integration between the UAS and human interface is the requirement to actually input specific GPS coordinates. This is a clunky system which needs to be inputted one-by-one—a bit of an excruciating experience. What's more, since the weather conditions can vary over long missions, and since it's often logical to have more than one drone in the sky, it made sense for the UAS to have limited knowledge of the exploration space and try to optimize its route based upon an area with key locations instead of all the necessary waypoints and Dubins (Chapter 2.3.3) alone.

One potential solution to this path problem is biologically-inspired. Ants leave their nest (takeoff) and investigate the surrounding area. When they discover a food source, they leave a particular trail of pheromones on their return.[66] As more ants explore the given grid, eventually the pheromones begin to collect along the quickest paths between the food sources and the nest. An ant colony was therefore implemented to represent the routing problem. A given ant can forage his environment and lay two types (food, nest) of pheromones (as well as sense the ones around him). The distance that the ant currently is from the source the weaker the pheromone will be, and the pheromones slowly evaporate over time.

An example of the outputs of this system can be seen in Figure 37. The colors represent pheromone trails from each food source and the black dots are

'ants' discovering the ideal path between the food. While this ant-based network was not implemented into the greater thesis project, if it were, it would be run in the background for approximately a thousand loops before providing a path for the drone and would be rerun each time conditions warrant. Nonetheless, the ants often jumped between path conclusions and, as can be seen in the figure, the path was not necessarily ideal. Therefore, this path methodology was abandoned before full implementation.

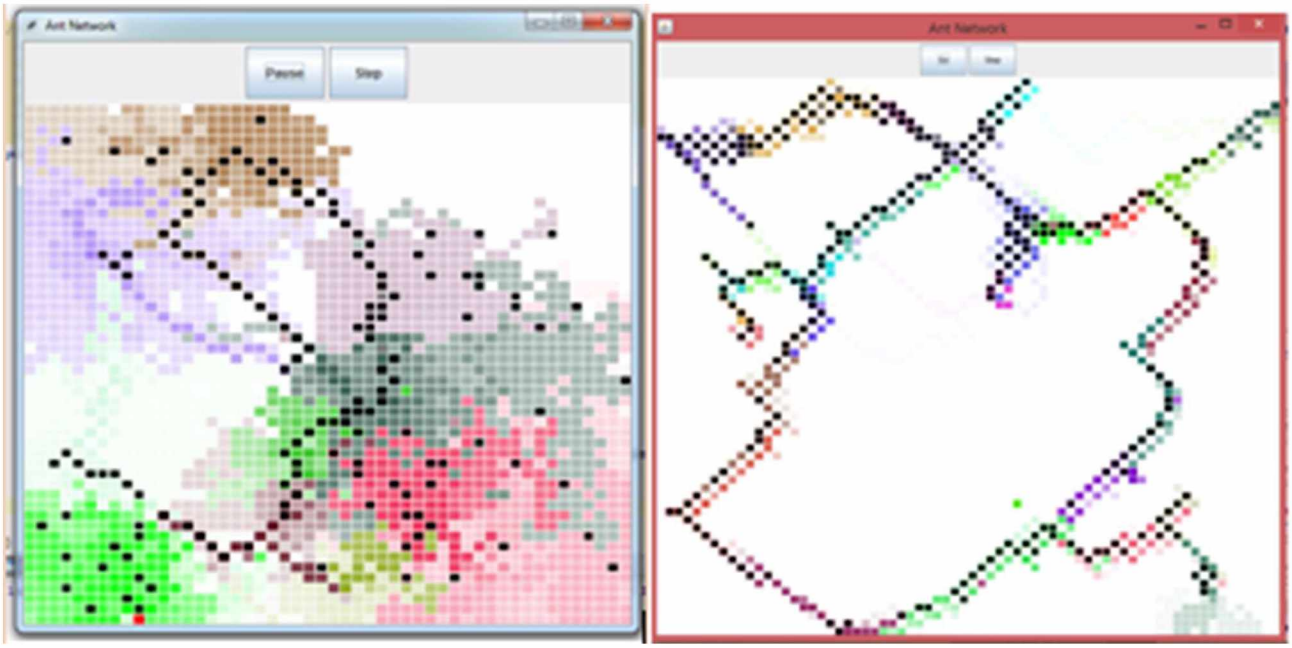


Figure 37. Example of ant-based path (left) during path discovery (right) after a solution convergence

6.4 Smartphone Limitations

One aspect of the smartphone that could lead to its replacement as the primary sensor source is the fact that its accelerometer and gyroscope sensors have

to be placed as close to the vehicle's center of gravity as possible. If these sensors were not at the center of gravity, then vehicle's rotations would also be perceived as accelerations, while vehicle's accelerations would also be perceived as rotations.

Since the center of gravity of the vehicle was often near the vehicle's center, this meant that the smartphone had to be embedded within the vehicle, making access somewhat difficult. One of the primary reasons for using a smartphone initially was its touchscreen which could not be accessed while the smartphone is within the configuration. If this feature were eliminated, the smartphone would thus become less desirable. The solution would then be to use alternate accelerometer and gyroscope sensors, and simply rely on the smartphone for its interface and camera capabilities. This is similar to many COTS systems, although it would allow the smartphone to be placed anywhere which would dramatically relax the design constraints of the vehicle.

Chapter 7: Conclusion

This paper detailed the completely autonomous capabilities of a UAS controlled by an onboard smartphone. This was solved by developing an Android application that could interface with a IOIO board for physical actuation. Given a cleared space about the size of a football field, GPS waypoint-driven missions, in conjunction with successful takeoffs and landings, were possible. Future endeavors look to leverage the vast capabilities of a smartphone, such as cellular communication, to modernize the UAS landscape.

7.1 Motivation and Summary

In Medieval European folklore, 'familiars' were entities that assisted in the activities of the 'cunning folk'.[67] This old folklore may just yet become a form of reality as the technology that is being carried around in the pockets of much of the world's population develops. The potential of smartphones, as new capabilities are added, is blossoming. This thesis imagines a future in which the everyday smartphone can be transfixed to any device and become its controller. The particular application of this thesis is upon drones, since the technology needed to essentially control a UAS has matured and been demonstrated.

To establish the magnitude represented herein by these pocket devices, a survey of the technological achievements in unmanned aircraft and control systems was completed in the first two chapters. This survey detailed the rich century and a half of history driven by wars and innovators and the significant efforts required to create a device which could keep itself aloft, particularly one that could complete an entire mission profile including waypoints, takeoff, and landing.

Chapter 3 discussed the technical backdrop of the primary thesis, specifically detailing what applicable control systems are, in particular the PID and why it relates to this thesis. Chapter 3 continues with an overview of the primary algorithm, often referred to as the fusion algorithm, which was developed to merge the sensor information into actionable data for flight. Lastly, Chapter 3 discussed the first step on the road to airborne autonomy, which was a ground vehicle successfully hitting waypoints and following a path.

Chapter 4 detailed the unmanned aerial system (UAS) which was used in this thesis to demonstrate smartphone control. The UAS consisted of a COTS Parkzone T-28 RC aircraft, with an embedded smartphone controller connected to a IOIO board for servo communication. Five channels were needed to control the servos of the elevator, ailerons, rudder, and speed controller.

In Chapter 5, the primary results of the UAS was outlined. The application developed on the smartphone was easy to use and visually confirm mission

changes as well as to send interrupt commands from a second device linked as a ground station. A full compilation video was posted online[62] that confirms smooth and complete control without human interference. Finally, Chapter 5 details how the non-RTOS limitation of the Android OS did not limit the capacity of the smartphone to reliably command the UAS.

Chapter 6 reviewed several additional explorations of capacity which were completed during the development of this thesis. This included a more mathematically rigorous methodology for determining the wind direction was investigated with the Unscented Kalman Filter and a LIDAR for more accurate landing. Academically, a biologically-inspired path following technique, given variadic starting conditions (ants), was additionally researched. Chapter 6 concluded with a discussion on why the smartphone is limited in its ability to replace embedded systems, particularly the constraints on placement and alignment near the center of gravity.

7.2 Key Results

The onboard Android smartphone successfully and autonomously controlled a drone through multiple complete flights including takeoff and landing. This thesis revealed that a smartphone can form the backbone of an entire UAS and thereby has the potential to replace the 'brain' of many robotic devices. Ergo, the

initial precepts of improving upon cost, usability, and utility were founded. The smartphone's downward pressure on embedded systems reduces the cost when compared to standard customized boards. The usability was inherent to the smartphone's clean user interface. The increased utility came in many forms, from the embedded sensors to the near plug-n-play sensor expansion capabilities of the heavily supported Android platform. As technology advances, with the billions invested in smartphones, these three categories only set to only widen the gap when compared to standard solutions. Especially as the cutting edge of voice, 3D capture, and AI features become ubiquitous.

Since the software was written as an application, another device was easily demonstrated to act as a ground station. The second device could send, via the XBee datalink, an interrupt command to alter the mission profile, or command a return home or loiter. This established the simplistic extensibility of a smartphone-based autopilot to coexist and communicate across multiple devices seamlessly.

The smartphone was an effective tool during the testing phase as it allowed quick access to programmatic adjustments that could be altered with the smartphone GUI directly; most UAS require manual code alterations at a desktop to perform similar adjustments. While non-RTOS was a concern to properly manage the dynamic control of an airborne vehicle, this thesis showed that even

under worst-case conditions, the Android OS was sufficient to achieve flight and complete an entire mission autonomously.

Lastly, a key differentiating feature of this thesis is its complete autonomy. Specifically, the ability for a smartphone app to drive takeoff, mission, and landing without human interruption. Most notably, the suite of connected and embedded sensors was sufficient to allow for curve following landing patterns within meters of the takeoff location that rival the author's experience with human capacity.

7.3 Future Work

This thesis established the potential of smartphones as the control for an unmanned aerial system, yet many pathways for extending the scope remain.

7.3.1 Simplified Design

This thesis uses both an external camera and barometer, mostly for convenience. Future work could eliminate both of these external devices as the camera quality on a smartphone more than surpasses that which was used here, and carefully crafted lenses could easily orient the picture frame at will. As previously discussed, some smartphones have quality barometers embedded, thus the external barometer could be easily removed. These two changes would simplify the external connections and overall design.

Currently, very few servos with USB-type connections exist. If custom servos were designed which could be networked into a single standard USB port, then the IOIO itself could theoretically be eliminated, requiring even fewer external devices to transition from smartphone to controller. These servos could even use Bluetooth or another wireless system for communication, though this would necessitate careful delay investigation as too much would result in an unstable system.

A smartphone already has several antennas which could be leveraged to remove the reliance on the XBee radios. At shorter distances, both WIFI and Bluetooth are potential options, but more powerfully, assuming the UAS is within the coverage area of a cell provider, the internet could be used to deliver interrupt commands from a ground station.

These three changes would greatly simplify the overall hardware diagram, essentially creating a smartphone with servos directly linked and no other necessary components. The author sees no major limitation to this, albeit the additional complexity for a few of the subsystems fell outside the scope of this thesis and was therefore skipped for expediency.

7.3.2 New Technology

3D vision is underway. Many of the large companies, including Microsoft, Facebook, Google, and more, are investing heavily in 3D technologies.[68] As these

technologies mature, they could be used directly to improve the UAS's capacity to sense and avoid obstacles. If, for example, a 3D capture technology were embedded within a smartphone, then it could be oriented in such a way as to improve object avoidance and overall performance.

Virtual and augmented reality glasses are also under heavy development. As these systems are synchronized with smartphones, such as the Samsung Galaxy[69], new forms of GUIs are on the horizon. The ability to quickly ascertain objects in three dimensions and then display them is potentially groundbreaking. The ubiquity of investment will certainly drive the market forward, and thereby improve the capacity for smartphones to command and control periphery devices.

NASA is currently researching prototype technologies to create a UAS Traffic Management (UTM) system that would permit safe operations of low-altitude vehicles; see Figure 38. One possible version of the UTM system takes advantage of cell towers. This would allow cooperative flight in densely populated areas for many UAS simultaneously. A smartphone is tailor-made to respond to such an environment as the necessary sensors are standard.

Figure 38. NASA proposed UTM system of low-altitude aircraft[70]

7.3.3 Entrepreneurial

Recently, there has been a dramatic rise in research and production of unmanned systems for both military and commercial use. Over the next decade, the UAS industry is expected to spend an estimated 12% of a total $98 billion on commercial applications.[18] Similar to the early computer development of the 80s, there is much room for developers to cast a unique vision and find success in this large and expanding market. As a result, the author of this thesis plans to leverage much of the experience learned along the way to develop an independent company that is focused on that proverbial crossroads of art, technology, and commerce, and builds upon the brilliant user experience that society expects of smartphones.

References

1. LeMieux, Jerry. Drone: UVA Dictionary. 1st ed. N.p.: CreateSpace Independent Platform, May 9, 2014. Print.
2. Kennett, Lee B. A History of Strategic Bombing. New York: Scribner, 1982. Chapter 6. Print.
3. Howeth, Captain Linwood S., USN (Retired), History of Communications-Electronics in the United States Navy, Washington, 1963, pgs 479-493, Print.
4. Hargrave, Lawrence. "Remote Piloted Aerial Vehicles." Remote Piloted Aerial Vehicles. Monash University, 2 Feb. 2003. Web. 28 Feb. 2016
5. Senior Airman, A.K. "RPAs Then and Now." Part III: History in the Making. Air Combat Command, 11 July 2013. Web. 28 Feb. 201
6. Fahmey, Delmar S., RADM USN "The Birth of Guided Missiles" United States Naval Institute Proceedings December 1980 pp 54-60
7. Spoto, Donald (2001). Marilyn Monroe: The Biography. Cooper Square Press. ISBN 978-0-8154-1183-3.
8. Shurcliff, Arthur A. Autobiography of Arthur A. Shurcliff: Written Winter 1943-1944, with Additions Summer of 1946, Summer of 1947, Houghton Library, p.9, Print.
9. Wagner, William. Lightning Bugs and Other Reconnaissance Drones. Washington: Armed Forces Journal, 1982. Print.
10. Dickson, Paul, and John Rothchild. The Electronic Battlefield: Wiring down the War. Washington, D.C.: Washington Monthly, 1971. Print.
11. Alexander, Grover L. "AQUILA Remotely Piloted Vehicle System Technology Demonstrator (RPV-STD) Program" Lockheed Missiles and Space Co, Final rept. Dec 1974-Dec 1977
12. "National Museum of the US Air Force™." Photos. Public Domain, 9 Oct. 1982. Web. 28 Feb. 2016.
13. Howard, Stephen P. Special Operations Forces and Unmanned Aerial Vehicles: Maxwell Air Force Base, Ala.: Air UP, 1996. Print.
14. Unclassified, United States " 0400: Research, Development, Test & Evaluation, Defense-Wide BA 7: Operational Systems Development." PE 0305219BB: MQ-1 Predator A UAV (Feb 2011): 1

15. Sweetman, Bill. "Aviation Week & Space Technology." 1986. Penton, 18 Aug. 2008. Web. 28 Feb. 2016.
16. United States. Office of Management and Budget. Budget FY 2016 - Department of Defense. N.p.: n.p., n.d. Budget FY 2016 - Department of Defense. 4 Feb. 2016.
17. Tackett, Everett. "Raytheon's Cobra Unmanned Aircraft System Makes Series of Firsts.": Investors: News Release. Raytheon, 11 July 2007. Web. 11 Aug. 2016.
18. Robbin, Ross. "DRONES: Quickly Navigating Toward Commercial Application, Starting with E-Commerce and Retail," BI Intelligence. Business Insider, 23 Jan. 2014. Web. 05 Feb. 2015.
19. Anderson, Chris. "ArduPilot - Arduino Compatible UAV Controller W/ ATMega328." - GPS-08785. Sparkfun, 20 Jan. 2010. Web. 28 Feb. 2016.
20. MacCready, Paul. "UAS: RQ-20A Puma AE." Puma AE UAS (UAV). AeroVironment, 8 June 2014. Web. 28 Feb. 2016.
21. Wang, Frank. "Phantom 3." Drone for Beginners. DJI, 10 Apr. 2015. Web. 28 Feb. 2016.
22. Goldman, Joshua. "Top-notch eye in the sky" DJI Phantom 2 Vision+ Review. CNET, 14 Oct. 2014. Web. 8 Feb. 2015
23. Avinc. "Puma AE." (n.d.) UAS: RQ-20A Puma AE. AeroVironment, 2014. Web. 8 Feb. 2015
24. Warren, Tom. "IPhone: A Visual History." The Verge. Vox Media, 09 Sept. 2014. Web. 28 Feb. 2016.
25. "Smartphone Sales Worldwide 2007-2015." Number of Smartphones Sold to End Users Worldwide from 2007 to 2015. Statista, Feb. 2016. Web. 11 Aug. 2016.
26. Tyler, Ryan., Alan, Fischer., Kim, H. Jin., "Design of an Attitude Rate Controller for Quadrotor Using an Android Smartphone," Seoul National University, doi: 10.2514/6.2012-4778, 13-16 August 2012 <https://code.google.com/p/quadrover/>
27. Desai, Alok., Lee, Dah-Jye., Moore, Jason., Chang, Yung-Ping., "Stabilization and Control of Quadrotor Helicopter Using a Smartphone Device", Proc. SPIE 8662, Intelligent Robots and Computer Vision, Algorithms and Techniques, 866208, doi:10.1117/ 12.2013703. 4 Feb. 2013

28. Thach, Do, Juhum, Kown, Chang-Joo Moon, "Ground System Software for Unmanned Aerial Vehicles on Android Device", World Academy of Science, Engineering and Technology, Vol 74, 27 Feb. 2013
29. Bissell, C. C. "Stodola, Hurwitz and the Genesis of the Stability Criterion." International Journal of Control 50.6 (1989): 2313-332. Web.
30. Tolle, Max. Die Regelung der Kraftmaschinen: Berechnung Und Konstruktion der Schwungr Der, Des. S.l.: Nabu, 2010. Print.
31. Mcruer, Duane, and Dunstan Graham. "Flight Control Century: Triumphs of the Systems Approach." Journal of Guidance, Control, and Dynamics 27.2 (2004): 161-73.
32. T.P. Hughes, Elmer Sperry: Inventor and Engineer. Baltimore, MD: Johns Hopkins Univ. Press, 1971. Print.
33. Bennett, Stuart. A History of Control Engineering: 1800-1930. New York: Peregrinus, 1979. Print.
34. Sweeten, Bill. "A Short History of Making Flying Safer." Photo Gallery: 100 Years of Aviation Safety History. AviationWeek, 15 Jan. 2016. Web. 04 Mar. 2016.
35. Bernstein, D. S. "Feedback control: an invisible thread in the history of technology," IEEE Control Systems, vol. 22, no. 2, pp. 53-68, Apr 2002.
36. Minorsky., N. "Directional Stability of Automatically Steered Bodies." Journal of the American Society for Naval Engineers 34.2 (1922): 280-309.
37. Bennett, S. "Nicholas Minorsky and the Automatic Steering of Ships." IEEE Control Syst. Mag. IEEE Control Systems Magazine 4.4 (1984): 10-15.
38. Bennett, Stuart. A History of Control Engineering: 1930-1955. London: Peregrinus, 1993. Print.
39. Mason. Stabilog. Foxboro, assignee. Patent 1897135. 1933. Print.
40. Oppelt, Winfried. "A Historical Review of Autopilot Development, Research, and Theory in Germany." J. Dyn. Sys., Meas., Control Journal of Dynamic Systems, Measurement, and Control 98.3 (1976): 215.
41. Bissell, C. C. "Karl Küpfmüller: A German Contributor to the Early Development of Linear Systems Theory." International Journal of Control 44.4 (1986): 977-89.
42. Nyquist, H. "Regeneration Theory." Bell System Technical Journal 11.1 (1932): 126-47.

43. Bode, Hendrik W. Network Analysis and Feedback Amplifier Design. New York: D. Van Nostrand, 1945. Print.
44. Kline, R. "Harold Black and the Negative-feedback Amplifier." IEEE Control Systems IEEE Control Syst. 13.4 (1993): 82-85.
45. Shinners, Stanley M. Modern Control System Theory and Design. New York: J. Wiley, 1992. Print.
46. Black, H. S. "Stabilized Feedback Amplifiers*." Bell System Technical Journal 13.1 (1934): 1-18.
47. Bennett, S. "The Emergence of a Discipline: Automatic Control 1940–1960." Automatica 12.2 (1976): 113-21.
48. Williams, T. J. "Computer Control Technology - Past, Present and Probable Future." Transactions of the Institute of Measurement and Control 5.1 (1983): 7-19.
49. Williams, T. J., Nof, S. Y. "Control Models," Handbook of Industrial Engineering, 2nd Ed., G. Salvendy, Ed., John Whiley and Sons, (1992): 211-238
50. Dubins, L. E. "On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents." American Journal of Mathematics 79.3 (1957): 497.
51. Lavelle, Steven M. "15.3.1 Dubins Curves." Dubins Curves., Cambridge University Press, 2006. Web. 13 Mar. 2016.
52. Kim, Jinsoo, Seongkyu Lee, Hoyong Ahn, Dongju Seo, Soyoung Park, and Chuluong Choi. "Feasibility of Employing a Smartphone as the Payload in a Photogrammetric UAV System." ISPRS Journal of Photogrammetry and Remote Sensing 79 (2013): 1-18.
53. M. Nagai, T. Chen, R. Shibasaki, H. Kumagai and A. Ahmed, "UAV-Borne 3-D Mapping System by Multisensor Integration," in IEEE Transactions on Geoscience and Remote Sensing, vol. 47, no. 3, pp. 701-708, March 2009.
54. Tabarracci, Joseph, and Patrick Currier. "A Blueprint for a Fixed-wing Autopilot on an Android Smartphone." 2013 Proceedings of IEEE Southeastcon (2013): n. pag.
55. Bronz, Murat. "UAV Autopilot." PaparazziUAV. École Nationale De L'aviation Civile (ENAC), 21 June 2016. Web. 11 Aug. 2016.
56. Ben-Tsvi, Ytai. "IOIO for Android." Sparkfun, Apr. 2011. Web. 11 Aug. 2016.

57. Desai, Alok, Dah-Jye Lee, Jason Moore, and Yung-Ping Chang. "Stabilization and Control of Quad-rotor Helicopter Using a Smartphone Device." Intelligent Robots and Computer Vision: Algorithms and Techniques (2013): n. pag.
58. Pachter, Meir, Nicola Ceccarelli, and Phillip Chandler. "Estimating MAV's Heading and the Wind Speed and Direction Using GPS, Inertial, and Air Speed Measurements." AIAA Guidance, Navigation and Control Conference and Exhibit (2008): n. pag.
59. W. Premerlani and P. Bizard, "Direction cosine matrix IMU: Theory,"DIY DRONE: USA, pp. 13–15, 2009. http://tinyurl.com/jduegay
60. Ben-Tsvi, Ytai. "Ytai/ioio." *GitHub*. Google, 10 Jan. 2014. Web. 13 Feb. 2015
61. Anderson, Chris. "GeoCrawler 2 (Cellphone Autopilot)." Geocrawler 2. DIY Drones, 12 June 2007. Web. 08 Feb. 2015
62. Gradwell, Greg., Bryant, Phillip., Claveau, David. "UAS Jan 2015 Flights." Flight Examples. <https://youtu.be/_o1R2SuQRBM> YouTube, 10 June 2015
63. Gradwell, Greg., Bryant, Phillip., Claveau, David. "UGV Path Following." Driving Examples. <https://youtu.be/sUI7Imn8NP4> YouTube, 12 June 2015
64. Pachter, Meir, Nicola Ceccarelli, and Phillip Chandler. "Estimating MAV's Heading and the Wind Speed and Direction Using GPS, Inertial, and Air Speed Measurements." AIAA Guidance, Navigation and Control Conference and Exhibit (2008): n. pag.
65. Lewis, Bob, and Dennis Corey. LIDAR. Digital image. Sparkfun Assets. N.p., n.d. Web. 10 Aug. 2016.
66. Simon. "Pheromone Trail Networks in Ants." Couzin Lab. Princeton University, 22 Sept. 2009. Web. 11 Aug. 2016.
67. Wilby, Emma. Cunning Folk and Familiar Spirits: Shamanistic Visionary Traditions in Early Modern British Witchcraft and Magic. Brighton: Sussex Academic, 2005. Print.
68. Blystone, Dan. "10 Stocks That Stand to Benefit From VR." Investopedia, 31 Dec. 2015. Web. 11 Aug. 2016.
69. Mistry, Pranav. "The Next-generation Gear VR Awaits." Samsung Electronics America. Samsung, 27 Nov. 2015. Web. 11 Aug. 2016.
70. Gilligan, Margaret, J. David Grizzle, and Victoria H. Cox. "Unmanned Aircraft System (UAS) Traffic Management (UTM)." UTM. NASA, 28 Sept. 2012. Web. 11 Oct. 2016.